

# Filter works

It's easy to filter works with the `filter` parameter:

- Get works where the publication year is 2020

[https://api.openalex.org/works?filter=publication\\_year:2020](https://api.openalex.org/works?filter=publication_year:2020) ↗

In this example the filter is `publication_year` and the value is 2020.

 It's best to [read about filters](#) before trying these out. It will show you how to combine filters and build an AND, OR, or negation query.

## /works attribute filters

You can filter using these attributes of the [Work](#) object (click each one to view their documentation on the [Work](#) object page):

 The `host_venue` and `alternate_host_venues` properties have been deprecated in favor of `primary_location` and `locations`. The attributes `host_venue` and `alternate_host_venues` are no longer available in the Work object, and trying to access them in filters or group-bys will return an error.

- `authorships.affiliations.institution_ids`
- `authorships.author.id` (alias: `author.id`) — Authors for a work (OpenAlex ID)
- `authorships.author.orcid` (alias: `author.orcid`) — Authors for a work (ORCID)
- `authorships.countries`
- `authorships.institutions.country_code` (alias: `institutions.country_code`)
- `authorships.institutions.id` (alias: `institutions.id`) — Institutions affiliated with the authors of a work (OpenAlex ID)
- `authorships.institutions.lineage`
- `authorships.institutions.ror` (alias: `institutions.ror`) — Institutions affiliated with the authors of a work (ROR ID)
- `authorships.institutions.type`
- `authorships.is_corresponding` (alias: `is_corresponding`) — This filter marks whether or not we have corresponding author information for a given work

- `apc_list.value`
- `apc_list.currency`
- `apc_list.provenance`
- `apc_list.value_usd`
- `apc_paid.value`
- `apc_paid.currency`
- `apc_paid.provenance`
- `apc_paid.value_usd`
- `best oa location.is_accepted`
- `best oa location.is_published`
- `best oa location.license` — The Open Access license for a work
- `best oa location.source.id`
- `best oa location.source.is_in_doaj`
- `best oa location.source.issn`
- `best oa location.source.host_organization`
- `best oa location.source.type`
- `best oa location.version`
- `biblio.first_page`
- `biblio.issue`
- `biblio.last_page`
- `biblio.volume`
- `cited_by_count`
- `concepts.id` (alias: `concept.id`) — The concepts associated with a work
- `concepts.wikidata`
- `corresponding_author_ids` — Corresponding authors for a work (OpenAlex ID)
- `corresponding_institution_ids`
- `countries_distinct_count`
- `doi` — The DOI (Digital Object Identifier) of a work
- `fulltext_origin`
- `fwci`
- `grants.award_id` — Award IDs for grants
- `grants.funder` — Funding organizations linked to grants for a work
- `has_content.pdf`
- `has_content.grobid_xml`
- `has_fulltext`

- `ids.pmcid`
- `ids.pmid` (alias: `pmid`)
- `ids.openalex` (alias: `openalex`) — The OpenAlex ID for a work
- `ids.mag` (alias: `mag`)
- `indexed_in`
- `institutions_distinct_count`
- `is_paratext`
- `is_retracted`
- `keywords.keyword`
- `language`
- `locations.is_accepted`
- `locations.is_oa`
- `locations.is_published`
- `locations.license`
- `locations.source.id`
- `locations.source.is_core`
- `locations.source.is_in_doaj`
- `locations.source.issn`
- `locations.source.host_organization`
- `locations.source.type`
- `locations.version`
- `locations_count`
- `open_access.any_repository_has_fulltext`
- `open_access.is_oa` (alias: `is_oa`) — Whether a work is Open Access
- `open_access. oa_status` (alias: `oa_status`) — The Open Access status for a work (e.g., gold, green, hybrid, etc.)
- `primary_location.is_accepted`
- `primary_location.is_oa`
- `primary_location.is_published`
- `primary_location.license`
- `primary_location.source.id`
- `primary_location.source.is_core`
- `primary_location.source.is_in_doaj`
- `primary_location.source.issn`
- `primary_location.source.host_organization`

- `primary_location.source.type`
- `primary_location.version`
- `primary_topic.id`
- `primary_topic.domain.id`
- `primary_topic.field.id`
- `primary_topic.subfield.id`
- `publication_year`
- `publication_date`
- `sustainable_development_goals.id`
- `topics.id`
- `topics.domain.id`
- `topics.field.id`
- `topics.subfield.id`
- `type`
- `type_crossref`

(i) Want to filter by the `display_name` of an associated entity (author, institution, source, etc.)? [See here.](#)

## /works convenience filters

These filters aren't attributes of the `Work` object, but they're handy for solving some common use cases:

### `abstract.search`

Text search using abstracts

Value: a search string

Returns: works whose abstract includes the given string. See the [search page](#) for details on the search algorithm used.

- Get works with abstracts that mention "artificial intelligence":

<https://api.openalex.org/works?filter=abstract.search:artificial%20intelligence> ↗

### `authors_count`

Number of authors for a work

Value: an Integer

Returns: works with the chosen number of `authorships` objects (authors). You can use the inequality filter to select a range, such as `authors_count:>5`.

- Get works that have exactly one author

[https://api.openalex.org/works?filter=authors\\_count:1](https://api.openalex.org/works?filter=authors_count:1) ↗

### `authorships.institutions.continent` (alias: `institutions.continent`)

Value: a String with a valid [continent filter](#)

Returns: works where at least *one* of the author's institutions is in the chosen continent.

- Get works where at least one author's institution in each work is located in Europe

<https://api.openalex.org/works?filter=authorships.institutions.continent:europe> ↗

### `authorships.institutions.is_global_south` (alias: `institutions.is_global_south`)

Value: a Boolean (`true` or `false`)

Returns: works where at least *one* of the author's institutions is in the Global South ([read more](#)).

- Get works where at least one author's institution is in the Global South

[https://api.openalex.org/works?filter=authorships.institutions.is\\_global\\_south:true](https://api.openalex.org/works?filter=authorships.institutions.is_global_south:true) ↗

### `best_open_version`

Value: a String with one of the following values:

- `any`: This means that `best_oa_location.version` = `submittedVersion`, `acceptedVersion`, or `publishedVersion`
- `acceptedOrPublished`: This means that `best_oa_location.version` can be `acceptedVersion` or `publishedVersion`
- `published`: This means that `best_oa_location.version` = `publishedVersion`

Returns: works that meet the above criteria for `best_oa_location`.

- Get works whose `best_oa_location` is a submitted, accepted, or published version:

[https://api.openalex.org/works?filter=best\\_open\\_version:any](https://api.openalex.org/works?filter=best_open_version:any) ↗

### `cited_by`

Value: the [OpenAlex ID](#) for a given work

Returns: works found in the given work's [referenced\\_works](#) section. You can think of this as **outgoing citations**.

- Get works cited by <https://openalex.org/W2766808518> ↗:  
[https://api.openalex.org/works?filter=cited\\_by:W2766808518](https://api.openalex.org/works?filter=cited_by:W2766808518) ↗

## cites

Value: the [OpenAlex ID](#) for a given work

Returns: works that cite the given work. This is works that have the given OpenAlex ID in the [referenced\\_works](#) section. You can think of this as **incoming citations**.

- Get works that cite <https://openalex.org/W2741809807> ↗:  
<https://api.openalex.org/works?filter=cites:W2741809807> ↗``

 The number of results returned by this filter may be slightly higher than the work's [cited\\_by\\_count](#) due to a timing lag in updating that field.

## concepts\_count

Value: an Integer

Returns: works with the chosen number of [concepts](#).

- Get works with at least three concepts assigned  
[https://api.openalex.org/works?filter=concepts\\\_count:>2](https://api.openalex.org/works?filter=concepts\_count:>2) ↗

## default.search

Text search across titles, abstracts, and full text of works

Value: a search string

This works the same as using the [search parameter](#) for Works.

## display\_name.search (alias: title.search)

Text search across titles for works

Value: a search string

Returns: works whose `display_name` (title) includes the given string; see the [search page](#) for details.

- Get works with titles that mention the word "wombat":

<https://api.openalex.org/works?filter=title.search:wombat> ↗

- ⓘ For most cases, you should use the `search` parameter instead of this filter, because it uses a better search algorithm and searches over abstracts as well as titles.

## from\_created\_date

Value: a date, formatted as `yyyy-mm-dd`

Returns: works with `created_date` greater than or equal to the given date.

This field requires an [OpenAlex Premium subscription to access. Click here to learn more.](#) ↗

- Get works created on or *after* January 12th, 2023 (does not work without valid API key):

[https://api.openalex.org/works?filter=from\\_created\\_date:2023-01-12&api\\_key=myapikey](https://api.openalex.org/works?filter=from_created_date:2023-01-12&api_key=myapikey) ↗

## from\_publication\_date

Value: a date, formatted as `yyyy-mm-dd`

Returns: works with `publication_date` greater than or equal to the given date.

- Get works published on or *after* March 14th, 2001:

[https://api.openalex.org/works?filter=from\\_publication\\_date:2001-03-14](https://api.openalex.org/works?filter=from_publication_date:2001-03-14) ↗

- ⓘ Filtering by publication date is not a reliable way to retrieve recently updated and created works, due to the way publishers assign publication dates. Use `from_created_date` or `from_updated_date` to get the latest changes in OpenAlex.

## from\_updated\_date

Value: a date, formatted as an [ISO 8601](#) ↗ date or date-time string (for example: "2020-05-17", "2020-05-17T15:30", or "2020-01-02T00:22:35.180390").

Returns: works with `updated_date` greater than or equal to the given date.

This field requires an [OpenAlex Premium subscription to access. Click here to learn more.](#) ↗

- Get works updated on or *after* January 12th, 2023 (does not work without valid API key):  
[https://api.openalex.org/works?filter=from\\_updated\\_date:2023-01-12&api\\_key=myapikey](https://api.openalex.org/works?filter=from_updated_date:2023-01-12&api_key=myapikey) ↗

ⓘ Learn more about using this filter to get the freshest data possible with our [Premium How-To](#) ↗.

## fulltext.search

Value: a search string

Returns: works whose fulltext includes the given string. Fulltext search is available for a subset of works, obtained either from PDFs or [n-grams](#), see [Work.has\\_fulltext](#) for more details.

- Get works with fulltext that mention "climate change": <https://api.openalex.org/works?filter=fulltext.search:climate%20change> ↗

ⓘ We combined some n-grams before storing them in our search database, so querying for an exact phrase using quotes does not always work well.

## has\_abstract

Works that have an abstract available

Value: a Boolean ([true](#) or [false](#))

Returns: works that have or lack an abstract, depending on the given value.

- Get the works that have abstracts:

[https://api.openalex.org/works?filter=has\\_abstract:true](https://api.openalex.org/works?filter=has_abstract:true) ↗

## has\_content.pdf

Value: a Boolean ([true](#) or [false](#))

Returns: works that have a downloadable PDF available via the [content endpoint](#).

- Get works with downloadable PDFs:

[https://api.openalex.org/works?filter=has\\_content.pdf:true](https://api.openalex.org/works?filter=has_content.pdf:true) ↗

## has\_content.grobid\_xml

Value: a Boolean ([true](#) or [false](#))

Returns: works that have downloadable TEI XML via the [content endpoint](#).

- Get works with downloadable TEI XML:

[https://api.openalex.org/works?filter=has\\_content.grobid\\_xml:true](https://api.openalex.org/works?filter=has_content.grobid_xml:true)

## has\_doi

Value: a Boolean (`true` or `false`)

Returns: works that have or lack a DOI, depending on the given value. It's especially useful for [grouping](#).

- Get the works that have no DOI assigned:

[https://api.openalex.org/works?filter=has\\_doi:false](https://api.openalex.org/works?filter=has_doi:false)

## has oa\_accepted\_or\_published\_version

Value: a Boolean (`true` or `false`)

Returns: works with at least one of the `locations` has `is_oa` = true and `version` is acceptedVersion or publishedVersion. For Works that undergo peer review, like journal articles, this means there is a peer-reviewed OA copy somewhere. For some items, like books, a published version doesn't imply peer review, so they aren't quite synonymous.

- Get works with an OA accepted or published copy

[https://api.openalex.org/works?filter=has\\_oa\\_accepted\\_or\\_published\\_version:true](https://api.openalex.org/works?filter=has_oa_accepted_or_published_version:true)

## has oa\_submitted\_version

Value: a Boolean (`true` or `false`)

Returns: works with at least one of the `locations` has `is_oa` = true and `version` is submittedVersion. This is useful for finding works with preprints deposited somewhere.

- Get works with an OA submitted copy:

[https://api.openalex.org/works?filter=has\\_oa\\_submitted\\_version:true](https://api.openalex.org/works?filter=has_oa_submitted_version:true)

## has\_orcid

Value: a Boolean (`true` or `false`)

Returns: if `true` it returns works where at least one author or has an [ORCID ID](#). If `false`, it returns works where no authors have an ORCID ID. This is based on the `orcid` field within `authorships.author`. Note that, sometimes, we assign ORCID using [author disambiguation](#), so this does not necessarily mean that the work itself has ORCID information.

- Get the works where at least one author has an ORCID ID:

[https://api.openalex.org/works?filter=has\\_orcid:true](https://api.openalex.org/works?filter=has_orcid:true) ↗

## has\_pmcid

Value: a Boolean (`true` or `false`)

Returns: works that have or lack a PubMed Central identifier (`pmcid`) depending on the given value.

- Get the works that have a `pmcid`:

[https://api.openalex.org/works?filter=has\\_pmcid:true](https://api.openalex.org/works?filter=has_pmcid:true) ↗

## has\_pmrid

Value: a Boolean (`true` or `false`)

Returns: works that have or lack a PubMed identifier (`pmid`), depending on the given value.

- Get the works that have a `pmid`:

[https://api.openalex.org/works?filter=has\\_pmrid:true](https://api.openalex.org/works?filter=has_pmrid:true) ↗

## has\_ngrams (DEPRECATED)

Works that have n-grams available to enable full-text search in OpenAlex.

This filter has been deprecated. See instead: [has\\_fulltext](#).

Value: a Boolean (`true` or `false`)

Returns: works for which n-grams are available or unavailable, depending on the given value. N-grams power fulltext searches through the `fulltext.search` filter and the `search` parameter.

- Get the works that have n-grams:

[https://api.openalex.org/works?filter=has\\_ngrams:true](https://api.openalex.org/works?filter=has_ngrams:true) ↗

## has\_references

Value: a Boolean (`true` or `false`)

Returns: works that have or lack `referenced_works`, depending on the given value.

- Get the works that have references:

[https://api.openalex.org/works?filter=has\\_references:true](https://api.openalex.org/works?filter=has_references:true) ↗

## journal

Value: the [OpenAlex ID](#) for a given [source](#), where the source is [type: journal](#)

Returns: works where the chosen [source ID](#) is the [primary\\_location.source](#).

### `locations.source.host_institution_lineage`

Value: the [OpenAlex ID](#) for an [Institution](#)

Returns: works where the given institution ID is in

### `locations.source.host_organization_lineage`

- Get the works that have <https://openalex.org/I205783295> in their `host_organization_lineage`:  
[https://api.openalex.org/works?  
filter=locations.source.host\\_institution\\_lineage:https://openalex.org/I205783295](https://api.openalex.org/works?filter=locations.source.host_institution_lineage:https://openalex.org/I205783295) ↗

### `locations.source.publisher_lineage`

Value: the [OpenAlex ID](#) for a [Publisher](#)

Returns: works where the given publisher ID is in

### `locations.source.host_organization_lineage`

- Get the works that have <https://openalex.org/P4310320547> in their `publisher_lineage`:  
[https://api.openalex.org/works?  
filter=locations.source.publisher\\_lineage:https://openalex.org/P4310320547](https://api.openalex.org/works?filter=locations.source.publisher_lineage:https://openalex.org/P4310320547) ↗

### `mag_only`

Value: a Boolean (`true` or `false`)

Returns: works which came from MAG (Microsoft Academic Graph), and no other data sources.

MAG was a project by Microsoft Research to catalog all of the scholarly content on the internet. After it was discontinued in 2021, OpenAlex built upon the data MAG had accumulated, connecting and expanding it using [a variety of other sources](#) ↗. The methods that MAG used to identify and aggregate scholarly content were quite different from most of our other sources, and so the content inherited from MAG, especially works that we did not connect with data from other sources, can look different from other works. While it's great to have these MAG-only works available, you may not always want to include them in your results or analyses. This filter allows you to include or exclude any works that came from MAG and only MAG.

- Get all MAG-only works:

[https://api.openalex.org/works?  
filter=mag\\_only:true](https://api.openalex.org/works?filter=mag_only:true) ↗

## primary\_location.source.has\_isbn

Value: a Boolean (`true` or `false`)

Returns: works where the `primary_location` has at least one ISBN assigned.

- Get the works that have an ISBN within the primary location:

[https://api.openalex.org/works?filter=primary\\_location.source.has\\_isbn:true](https://api.openalex.org/works?filter=primary_location.source.has_isbn:true) ↗

## primary\_location.source.publisher\_lineage

Value: the [OpenAlex ID](#) for a [Publisher](#)

Returns: works where the given publisher ID is in

## primary\_location.source.host\_organization\_lineage

- Get the works that have <https://openalex.org/P4310320547> in their `publisher_lineage`:

[https://api.openalex.org/works?filter=primary\\_location.source.publisher\\_lineage:https://openalex.org/P4310320547](https://api.openalex.org/works?filter=primary_location.source.publisher_lineage:https://openalex.org/P4310320547) ↗

## raw\_affiliation\_strings.search

(i) This filter used to be named `raw_affiliation_string.search`, but it is now `raw_affiliation_strings.search` (i.e., plural, with an 's').

Value: a search string

Returns: works that have at least one `raw_affiliation_strings` which includes the given string. See the [search page](#) for details on the search algorithm used.

- Get works with the words *Department of Political Science, University of Amsterdam* somewhere in at least one author's `raw_affiliation_strings`:

[https://api.openalex.org/works?filter=raw\\_affiliation\\_strings.search:department%20of%20political%20science%20university%20of%amsterdam](https://api.openalex.org/works?filter=raw_affiliation_strings.search:department%20of%20political%20science%20university%20of%amsterdam) ↗

## related\_to

Value: the [OpenAlex ID](#) for a given work

Returns: works found in the given work's `related_works` section.

- Get works related to <https://openalex.org/W2486144666> ↗:

[https://api.openalex.org/works?filter=related\\_to:W2486144666](https://api.openalex.org/works?filter=related_to:W2486144666) ↗

## repository

Value: the [OpenAlex ID](#) for a given [source](#), where the source is [type: repository](#)

Returns: works where the chosen [source ID](#) exists within the [locations](#).

You can use this to find works where authors are associated with your university, but the work is not part of the university's repository. 🤝

- Get works that are available in the University of Michigan Deep Blue repository (OpenAlex ID: <https://openalex.org/S4306400393>)  
<https://api.openalex.org/works?filter=repository:S4306400393> ↗
- Get works where at least one author is associated with the University of Michigan, but the works are not found in the University of Michigan Deep Blue repository  
<https://api.openalex.org/works?filter=institutions.id:I27837315,repository:!S4306400393> ↗

You can also use this as a [group\\_by](#) to learn things about repositories:

- Learn which repositories have the most open access works  
[https://api.openalex.org/works?filter=is\\_oa:true&group\\_by=repository](https://api.openalex.org/works?filter=is_oa:true&group_by=repository) ↗

## title\_and\_abstract.search

Text search across titles and abstracts for works

Value: a search string

Returns: works whose [display\\_name](#) (title) or abstract includes the given string; see the [search page](#) for details.

- Get works with title or abstract mentioning "gum disease":

[https://api.openalex.org/works?filter=title\\_and\\_abstract.search:gum%20disease](https://api.openalex.org/works?filter=title_and_abstract.search:gum%20disease) ↗

## to\_created\_date

Value: a date, formatted as [yyyy-mm-dd](#)

Returns: works with [created\\_date](#) less than or equal to the given date.

This field requires an [OpenAlex Premium subscription](#) to access. [Click here to learn more.](#) ↗

- Get works created on or *after* January 12th, 2023 (does not work without valid API key):  
[https://api.openalex.org/works?filter=to\\_created\\_date:2024-01-12&api\\_key=myapikey](https://api.openalex.org/works?filter=to_created_date:2024-01-12&api_key=myapikey) ↗

## to\_publication\_date

Value: a date, formatted as `yyyy-mm-dd`

Returns: works with `publication_date` less than or equal to the given date.

- Get works published on or *before* March 14th, 2001:

[https://api.openalex.org/works?filter=to\\_publication\\_date:2001-03-14](https://api.openalex.org/works?filter=to_publication_date:2001-03-14) ↗

## to\_updated\_date

Value: a date, formatted as an [ISO 8601](#) ↗ date or date-time string (for example: "2020-05-17", "2020-05-17T15:30", or "2020-01-02T00:22:35.180390").

Returns: works with `updated_date` less than or equal to the given date.

This field requires an [OpenAlex Premium subscription](#) to access. [Click here to learn more.](#) ↗

- Get works updated before or on January 12th, 2023 (does not work without valid API key):

[https://api.openalex.org/works?filter=to\\_updated\\_date:2023-01-12&api\\_key=myapikey](https://api.openalex.org/works?filter=to_updated_date:2023-01-12&api_key=myapikey) ↗

## version

Value: a String with value `publishedVersion`, `submittedVersion`, `acceptedVersion`, or `null`

Returns: works where the chosen version exists within the `locations`. If `null`, it returns works where no version is found in any of the locations.

- Get works where a published version is available in at least one of the locations:

<https://api.openalex.org/works?filter=version:publishedVersion> ↗



Previous

Get lists of works



Next

Search works

Last updated 3 days ago