Scopus Search API

This represents the search interfaces associated with Scopus abstracts. Each search result will, by definition, link to a Scopus abstract (https://dev.elsevier.com/documentation/AbstractRetrievalAPI.wadl). Search entries might also have links to a Full-Text article (https://dev.elsevier.com/documentation/ArticleRetrievalAPI.wadl) representation of the result.

Additional information includes Scopus Search Views (https://dev.elsevier.com/guides/ScopusSearchViews.htm) and Scopus Search Tips (https://dev.elsevier.com/tips/ScopusSearchTips.htm).

Summary

Resource	Method	Description
https://api.elsevier.com/content/search/scopus	GET	SCOPUS Search API: This represents a search against the SCOPUS cluster, which contains SCOPUS abstracts. Scopus is the largest abstract and citation database of research literature and quality web sources. Updated daily, Scopus covers 50 million abstracts of over 20,500 peer-reviewed titles from more than 5,000 publishers. This search resource allows for the submission of Boolean queries into the Scopus index, retrieving relevant result metadata in a user-specific text formats.

Resources

https://api.elsevier.com/content/search/scopus

SCOPUS Search API: This represents a search against the SCOPUS cluster, which contains SCOPUS abstracts. Scopus is the largest abstract and citation database of research literature and quality web sources. Updated daily, Scopus covers 50 million abstracts of over 20,500 peer-reviewed titles from more than 5,000 publishers. This search resource allows for the submission of Boolean queries into the Scopus index, retrieving relevant result metadata in a user-specific text formats.

Methods

GET simple()

request

header params

Accept	xsd:string (required) default: `application/json` options: `application/json`, `application/atom+xml`, `application/xml`	This represents the acceptable mime type format in which the response can be generated. This can also be submitted as the query string parameter "httpAccept". This returns the response in JSON, ATOM, or XML mark-up.
Authorization	xsd:string	This header field contains the OAuth bearer access token in which the format of the field is "Bearer <token>" (where the token represents the end-user session key). The presence of a bearer token implies the request will be executed against user-based entitlements. The Authorization field overrides X-ELS-Authtoken.
X-ELS-APIKey	xsd:string (required)	This represents a unique application developer key providing access to API resources. This key can also be submitted as the query string parameter "apiKey"
X-ELS-Authtoken	xsd:string	This represents a end-user session. If provided, this token is used to validate the credentials needed to access content in this resource. This token can also be submitted through the HTTP header "Authorization" or the query string parameter "access_token".
X-ELS-Insttoken	xsd:string	This represents a institution token. If provided, this key (in combination with its associated APIKey) is used to establish the credentials needed to access content in this resource. This token can also be provided through the query string parameter "insttoken".
X-ELS-ReqId	xsd:string	This is a client-defined request identifier, which will be logged in all trace messages of the service. This identifier can be used to track a specific transaction in the service's message logs. It will also be returned as an HTTP header in the corresponding response. Note that this should be a unique identifier for the client, and used to track a single transaction.

X-ELS-ResourceVersion	xsd:string options: `facetexpand`, `allexpand`, `new`	Represents the version of the resource that should be received. Multiple attributes can be submitted by separating with commas or semicolons. Options include: **facetexpand** - adds new fields under each facet returned (where applicable) **allexpand** - (same as facetexpand) **new** - returns the most recent and prototyped features

query params

httpAccept	xsd:string options: `application/json`, `application/atom+xml`, `application/xml`	Override for HTTP header Accept, this represents the acceptable mime types in which the response can be generated.
access_token	xsd:string	Override for HTTP header Authorization, this contains the OAuth bearer access token, where the format of the field is "<token>" (where the token represents the end-user session key). The presence of a bearer token implies the request will be executed against user-based entitlements.
insttoken	xsd:string	Override for HTTP header X-ELS-Insttoken, this represents a institution token. If provided, this key (in combination with its associated APIKey) is used to establish the credentials needed to access content in this resource.
apiKey	xsd:string	Override for HTTP header X-ELS-APIKey, this represents a unique application developer key providing access to resource.
reqId	xsd:string	Override for HTTP header X-ELS-ReqId, this is a client-defined request identifier, which will be logged in all trace messages of the service. This identifier can be used to track a specific transaction in the service's message logs. Note that this should be a unique identifier for the client, and used to track a single transaction.
ver	xsd:string options: `facetexpand`, `allexpand`, `new`	Override for HTTP header X-ELS-ResourceVersion, this represents the version of the resource that should be received. Multiple attributes can be submitted by separating with commas or semicolons. Options include: **facetexpand** - adds new fields under each facet returned (where applicable) **allexpand** - (same as facetexpand) **new** - returns the most recent and prototyped features
query	xsd:string (required)	This represents the boolean search to be executed against the SCOPUS cluster. There is additional information regarding Search Tips (https://dev.elsevier.com/tips/ScopusSearchTips.htm). ex. query=heart+attack%20AND%20text(liver)
view	xsd:string default: `STANDARD` options: `STANDARD`, `COMPLETE`	This alias represents the list of elements that will be returned in the response. The following chart shows the Scopus Search Views (https://dev.elsevier.com/guides/ScopusSearchViews.htm).

field	xsd:string	This alias represents the name of specific fields that should be returned. The list of fields include all of the fields returned in the response payload (see view). Multiple fields can be specified, delimited by commas. Note that specifying this parameter overrides the view parameter. ex. field=url,identifier,description
suppressNavLinks	xsd:string default: `false` options: `true`, `false`	This parameter is used to suppress the inclusion of top-level navigation links in the response payload.
date	xsd:string	Represents the date range associated with the search, with the lowest granularity being year. ex. date=2002-2007
start	xsd:string	Numeric value representing the results offset (i.e. starting position for the search results). The maximum for this value is a system-level default (varies with search cluster) minus the number of results requested. If not specified the offset will be set to zero (i.e. first search result) ex. start=5
count	xsd:string	Numeric value representing the maximum number of results to be returned for the search. If not provided this will be set to a system default based on service level. In addition the number cannot exceed the maximum system default - if it does an error will be returned. ex. count=10
sort	xsd:string options: `artnum`, `citedby-count`, `coverDate`, `creator`, `orig-load-date`, `pagecount`, `pagefirst`, `pageRange`, `publicationName`, `pubyear`, `relevancy`, `volume`	Represents the sort field name and order. A plus in front of the sort field name indicates ascending order, a minus indicates descending order. If sort order is not specified (i.e. no + or -) then the order defaults to descending (DESC). Up to three fields can be specified, each delimited by a comma. The precedence is determined by their order (i.e. first is primary, second is secondary, and third is tertiary). +/-{field name}[,+/-{field name} ex. sort=coverDate,-title
content	xsd:string default: `all` options: `core`, `dummy`, `all`	This parameter is used to filter specific categories of content that should be searched/returned.

subj	xsd:string	Represents the subject area code associated with the content category desired. Note that these subject code mapping vary based upon the environment in which the request is executed. The following resource shows all of the available subject classifications (https://api.elsevier.com/content/subject/scopus?httpAccept=text/xml). Available choices include: **AGRI** - Agricultural and Biological Sciences **ARTS** - Arts and Humanities **BIOC** - Biochemistry, Genetics and Molecular Biology **BUSI** - Business, Management and Accounting **CENG** - Chemical Engineering **CHEM** - Chemistry **COMP** - Computer Science **DECI** - Decision Sciences **DENT** - Dentistry **EART** - Earth and Planetary Sciences **ECON** - Economics, Econometrics and Finance **ENER** - Energy **ENGI** - Engineering **ENVI** - Environmental Science **HEAL** - Health Professions **IMMU** - Immunology and Microbiology **MATE** - Materials Science **MATH** - Mathematics **MEDI** - Medicine **NEUR** - Neuroscience **NURS** - Nursing **PHAR** - Pharmacology, Toxicology and Pharmaceutics **PHYS** - Physics and Astronomy **PSYC** - Psychology **SOCI** - Social Sciences **VETE** - Veterinary **MULT** - Multidisciplinary

alias	xsd:string default: `true` options: `true`, `false`	This parameter controls the default behavior of returning a superseded author profiles. Submitting this parameter as false will override the default behavior. This is only applicable for searches targeting author identifiers.
cursor	xsd:string	This parameter is used when a user wants to execute deep pagination searching (i.e. iterate to the end of a search result set). Under normal circumstances, when using the 'start' parameter (results offset), access to the total result set is limited to a predefined maximum number of results. By using the cursor in place of the 'start' the user can iterate to the very end of the result set, with the restriction that results can only be accessed by iterating forward sequentially (there will be no 'prev' or 'last' links available). This capability is initially accessed by sending a "*" in the first search request. Subsequent requests should submit the 'cursor/@next' value from each corresponding response as the 'cursor' value. The 'cursor/@next' value must be URL encoded by the client application. The navigation links ('next') can also be used to navigate to each succeeding search result entry and are URL encoded by default.

facets	xsd:string options: `af-id`, `aucite`, `au-id`, `authname`, `country`, `exactsrctitle`, `fund-sponsor`, `language`, `openaccess`, `pubyear`, `restype`, `srctype`, `subjarea`	Represents the navigator that should be included in the search results. One or more navigators can be specified on the request, delimited by a semicolon Different dimensions of the navigator will be represented within parentheses. These dimensions include: count : the number of "buckets" to include (i.e. how many navigator entries) sort : how the navigators should be sorted. Options include na (Modifier name, ascending), fd (Modifier frequency, descending), and fdna (Modifier frequency descending, secondary sort through unity by name, ascending). prefix : filters the facet values to only those matching the prefix specified (not applicable for numeric values). In addition, qualifiers can be specified to include or exclude values from the response (i.e. filter). The options: include : include only the value specified from the facet response. include_above : include only values above the value specified from the facet response. include_below : include only values below the value specified from the facet response. exclude : exclude only the value specified from the facet response. exclude_above : exclude all values above the value specified from the facet response. exclude_below : exclude all values below the value specified from the facet response. ex. facets=authname(count=20,sort=na,prefix=Ma);exactsrctitle (prefix=J);subjarea(sort=fd);pubyear;srctype(sort=fdna) Available facets include: **AF-ID** - affiliation identifier **AUCITE** - author citation **AU-ID** - author identifier **AUTHNAME** - author identifier and author name **COUNTRY** - affiliation country **EXACTSRCTITLE** - source title **FUND-SPONSOR** - funding sponsor **LANGUAGE** - language **OPENACCESS** - open access status **PUBYEAR** - publication year **RESTYPE** - internal collection **SUBJAREA** - subject area **SRCTYPE** - content category

responses

status:
200

representations

application/json	SCOPUS JSON Response: This is XML representing the requested document.	JSON Example (https://dev.elsevier.com/payloads/search/scopusSearchResp.json)

status:
200

representations

application/atom+xml	SCOPUS ATOM Response: This is XML representing the requested document.	ATOM+XML Example (https://dev.elsevier.com/payloads/search/scopusSearchATOMResp.xml)

status:
200

representations

application/xml	SCOPUS XML Response: This is XML representing the requested document.	XML Example (https://dev.elsevier.com/payloads/search/scopusSearchResp.xml)

status:
400

representations

text/xml	Invalid Request: This is an error that occurs when invalid information is submitted.	Invalid Request Example (https://dev.elsevier.com/payloads/invalidRequest.xml)

status:
401

representations

text/xml	Authentication Error: This is an error that occurs when a user cannot be authenticated due to missing/invalid credentials (authtoken or APIKey).	Authentication Error Example (https://dev.elsevier.com/payloads/authError.xml)

status:
403

representations

text/xml	Authorization/Entitlements Error: This is an error that occurs when a user cannot be authenticated or entitlements cannot be validated.	Authorization Error Example (https://dev.elsevier.com/payloads/authError.xml)

status:
405

representations

text/xml	Invalid HTTP Method: This is an error that occurs when the requested HTTP Method is invalid.	Invalid Method Example (https://dev.elsevier.com/payloads/invalidHttpMethod.xml)

status:
406

representations

text/xml	Invalid Mime Type: This is an error that occurs when the requested mime type is invalid.	Invalid Mime Type Example (https://dev.elsevier.com/payloads/invalidAcceptHeader.xml)

status:
429

representations

text/xml	Quota Exceeded: This is an error that occurs when a requester has exceeded the quota limits associated with their API Key.	Quota Exceeded Example (https://dev.elsevier.com/payloads/quotaExceeded.xml)

status:
500

representations

text/xml	Generic Error: This is a general purpose error condition, typically due to back-end processing errors.	Generic Error Example (https://dev.elsevier.com/payloads/genericError.xml)