OpenSearch API

From May 1, 2013, Nature Publishing Group will remove the requirement for an API key to make calls to our open APIs. The APIs for blog listings and OpenSearch will continue to be found at the api.nature.com endpoint. If you are currently calling the API with an API key, **you do not need to make any changes to your code**, but if you are writing new code, you will no longer need to acquire and pass a key as a parameter. If you have any questions, please feel free to send us an email at developers@nature.com

Nature.com OpenSearch

The nature.com OpenSearch API provides an open, bibliographic search service for content hosted on nature.com, comprising around half a million news and research articles and citations.

Indexes for all journals listed at nature.com are searchable. Indexes cover the complete publication history for each journal from date of first publication, which in some cases extends back more than 150 years.

Access to the OpenSearch API requires an API Key and is subject to the API Terms of Use.

Quick Links: API Overview | API Reference

Search Interfaces

The OpenSearch API presents two interfaces:

- a simple OpenSearch API based on the OpenSearch industry standard, used where a keyword search is sufficient, typically returning results as an RSS feed
- a full SRU search interface, supporting structured queries using the CQL query language, returning structured results with an XML schema, as well as a full diagnostics set. Alternate result formats are available through content negotiation and/or by query string parameter

See More about the SRU Protocol and More about CQL, below.

Note: you may also find the our cheatsheet **(pdf download) helpful.**

For an insight into how the nature.com implementation combines these two search approaches into a single service, see this project Case Study.

API Overview

The OpenSearch API is a REST API that returns search results for HTTP queries.

Multiple result formats are supported, including RSS, JSON, and ATOM, as well as SRU (default) XML format. Support for common, simple, list-based result formats makes the service particularly suitable for AJAX-style integration into web-based apps and dynamic web pages, as well as more traditional user-directed searches.

A developer key will ultimately be required to call the OpenSearch API, but it is not currently needed for new developers assessing the API or its existing callers. The transition to a key-based API will be gradual, and communicated in advance by Nature Publishing Group in the latter half of 2011.

The base URL and service endpoint for nature.com OpenSearch is http://api.nature.com/content/opensearch

Searches are constructed as query strings.

The simplest type of search is keyword search for a single keyword, for example:

```
http://api.nature.com/content/opensearch
query=darwin1&api_key=<API key string here
```

returning records that include references to Darwin.

Simple keyword search for a list of keywords is supported using the queryTerms parameter, for example:

```
http://api.nature.com/content/opensearch
queryType=searchTerms&query=darwin+lamarck
here>
```

returning records that include references to Darwin *and* Lamarck; in keyword search, the list of terms is treated as a Boolean AND expression.

Note that whitespace will be rejected in query strings; instead use “+” as the space character.

For more complex searches, OpenSearch implements the SRU search protocol supporting CQL query syntax. For example, this CQL query is a Boolean OR search for documents containing either term, Darwin *or* Lamarck:

```
http://api.nature.com/content/opensearch
queryType=cql&query=cql.keywords+any+darwi
ck&api_key=<API key string here>
```

To find out more about CQL syntax, see More about CQL, below.

The following standard formats are supported for search results, and are specified in the query string with the httpAccept parameter:

- ATOM
- JSON
- JSONP (JSON with padding)
- RSS
- SRU (default)
- TURTLE

For example, to return JSON format results for the above query:

```
http://api.nature.com/content/opensearch
queryType=cql&query=cql.keywords+any+darwi
ck&httpAccept=application/json&api_key=<AP
```

For complete details, see the API Reference below.

Result sorting orders can be specified based on relevance, oldest/newest, article type, title, or journal name.

Result sets are identified with a ‘resultSetId’ which has a lifetime of 1 h (3600 s). Note that in this implementation the ‘resultSetId’ is used to identify the original query string itself rather than the actual result set records as per a strict interpretation of the SRU protocol. In general practice this will yield consistent results although users should note that there is a small possibility of new records being added to the service between subsequent retrievals.

Results sets from the nature.com OpenSearch service are returned in relevance order by default and are served up in pages of 25 records. Additional records can be retrieved by paging through the result set. The page size can be varied and is capped at 100 records.

More about the SRU Protocol

SRU stands for Search/Retrieve via URL and defines a full protocol for communicating with a search engine.

SRU query string parameters define both the content of the search, and how the search should be performed:

- The query parameter holds the query clause that defines the search content, which is specified using CQL syntax
- The remaining parameters are control parameters that determine how the server should conduct the search

The SRU specification is currently being standardised through OASIS, the Information Standards consortium.

More about CQL

The query clause of a SRU request is constructed using CQL, Contextual Query Language, which defines a complete and formal query language and is not just limited to bibliographic information. As the spec says “CQL is a formal language for representing queries to information retrieval systems such as web indexes, bibliographic catalogs and museum collection information.”

For a simple summary, see for example A Gentle Introduction to CQL.

Like SRU, CQL is also currently being standardised through OASIS.

API Reference

API Summary

This table summarises OpenSearch API request parameters. For full details of usage including examples, see the individual entries in the API Description.

Name:	OpenSearch
Endpoint:	http://api.nature.com/content/opensearch/request

/request	**Required:** • query **Optional:** • facetLimit, httpAccept, maximumRecords, operation,

	queryType, recordPacking, recordSchema, sortKeys, startRecord, version **Not supported:** • extraRequestData, recordXPath, resultSetTTL, stylesheet

API Description

Endpoint

The endpoint for the OpenSearch API is the URL:

- http://api.nature.com/content/opensearch/request/

Access, Rate Limiting, Security, and Performance

Result formats

By default, results are returned in SRU XML format. The following formats are supported, and can be set as required in the query string. For details, see httpAccept below:

- ATOM
- JSON
- JSONP (JSON with padding)
- RSS
- SRU (default)
- TURTLE

Errors

Errors are returned as standard HTTP errors. For example, typically, “HTTP Error 400: Bad Request” is returned for an ill-formed query; “HTTP Error 500: Internal server error” indicates a server error.

Parameters

Required parameters

The following parameters are always required:

Name:	query
Value:	String, a term or multiple terms for simple keyword search, or a full CQL syntax query expression**Note:** By default, CQL search is assumed for all queries except

	single keyword queries. For simple keyword search against multiple terms, add the queryType parameter set to searchTerms**Note:** Use “+” in all query types as a substitute for whitespace

Notes:

In a keyword search against a list of terms, the queryType parameter is required, set to searchTerms, and the search terms are treated as an ANDed list.

For more complex searches, set query to a well-formed CQL query clause constructed from a combination of terms conforming to CQL syntax.

The query parameter is always required.

Examples

- Simple keyword search with a single keyword:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch?
queryType=searchTerms&query=darwin&api_key
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch?
Terms%26query%3Ddarwin%26api_key=<API key
```

- Simple keyword search with multiple keywords, performs ANDed search:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch?
queryType=searchTerms&query=darwin+lamarck
here>
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch?
Terms%26query%3Ddarwin%2Blamarck%26api_key
```

- CQL query syntax, performs ORed search:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch?
queryType=cql&query=cql.keywords+any+darwi
ck&api_key=<API key string here>
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch?
query%3Dcql.keywords%2Bany%2Bdarwin%2BOR%2
ck%26api_key=<API key string here>
```

Optional parameters

The following optional parameters are supported.

Name:	facetLimit
Value:	A number, the maximum item count for a query “facet”

Notes

SRU-defined parameter. Specifies the maximum number of items to return for matches against a CQL search “facet”, for example dc.title for a journal title search. The first record is indexed 0, hence a facetLimit of 2 will return at most 3 records.

Examples

- CQL search for titles including Darwin AND Lamarck, for example “Evolution pioneers: celebrating Lamarck at 200, Darwin 215”, “Darwin Versus Lamarck”, etc. Returns 3 records:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch?
queryType=cql&query=dc.title+any+darwin+AN
tLimit:dc.title=2&api_key=<API key string
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch?
query%3Ddc.title%2Bany%2Bdarwin%2BAND%2Bdc
cetLimit%3Adc.title%3D2%26api_key=<API key
```

Name:	httpAccept
Value:	A supported media type, one of those from the list below, specified as indicated

Notes

SRU-defined parameter. By default, results are returned in SRU XML format. To specify an alternative format, include the httpAccept parameter in the query string, set to a supported media type.

Supported media types:

Requested format	Media type
HTML	text/html
XML	application/xml
SRU	application/sru+xml
RSS	application/rss+xml

ATOM	application/atom+xml
JSON	application/json
JSONP	text/javascript

Example

- Request search results in JSON format:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch
queryType=cql&query=cql.keywords+any+darwi
ck&httpAccept=application/json&api_key=<AP
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch
query%3Dcql.keywords%2Bany%2Bdarwin%2BOR%2
ck%26httpAccept%3Dapplication/json%26api_k
```

Name:	maximumRecords
Value:	Number, 0 or greater

Notes

SRU-defined parameter. Sets the number of records to be returned. If this parameter is not supplied, it will default to X.

The parameter defines the maximum number of records that should be returned. If fewer records match the search criteria, then a smaller number of documents will be returned, i.e. those that match the search.

For mobile applications, or other cases where bandwidth or system resources may be limited, setting this value can help you tune your app performance. Explicitly setting a very large number may impact server performance and lead to slower responses.

Example

- Limit the number of records returned for this search to 5:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch
queryType=searchTerms&query=darwin+lamarck
<API key string here>
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch
Terms%26query%3Ddarwin%2Blamarck%26maximum
key string here>
```

Name:	operation

Value:	Legacy parameter

Notes

Legacy, defined by the SRU standard. While supported, this parameter is intended only for use by legacy applications.

Name:	queryType
Value:	A supported query type, one of those from the list below, specified as indicated

Notes

Specify how the server should interpret the query string. By default, queries are interpreted as:

- queryType not specified, query= a single keyword, treated as a simple keyword search
- queryType not specified, query= multiple query terms, treated as a CQL expression and will be rejected if not well-formed

To specify a simple keyword search against multiple keywords, set queryType to searchTerms; the terms ("+" or "," separated) will be treated as an ANDed list of terms to be matched.

For CQL queries, setting queryType to cql is equivalent to the default behaviour.

Note that this parameter is NOT part of the SRU standard.

Possible values are:

Query type	**Meaning**
searchTerms	Treat the query string as an ANDed list of query terms
cql	Treat the query string as a CQL query clause

Example

- Perform a keyword search against multiple keywords:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch?
queryType=searchTerms&query=darwin+lamarck
here>
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch?
```

```
Terms%26query%3Ddarwin%2Blamarck%26api_key
```

- Perform a CQL search:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch?
queryType=cql&query=cql.keywords+any+darwi
ck&api_key=<API key string here>
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch?
query%3Dcql.keywords%2Bany%2Bdarwin%2BOR%2
ck%26api_key=<API key string here>
```

Name:	recordPacking
Value:	A supported packing type, see below

Notes

SRU-defined parameter. Determines how the record should be escaped in the response. The default packing is “xml”.

Typically, if your app will process the result, for example to parse fields from the record returned, then specify “string” packing; results will be returned with special characters escaped. However, if you expect to present the result directly to users within your app page, for example applying stylesheets or transformations directly to the result, then specify “xml”, or omit the parameter to enable xml by default.

Possible values are:

Packing type	**Meaning**
string	Return a string result suitable e.g. for application parsing
xml	Return an XML format result suitable e.g. for direct presentation to users via stylesheet
unpacked	??

Example

- Request JSON results, unpacked:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch?
queryType=cql&query=dc.title+any+darwin+AN
Keys=title&httpAccept=application/json&rec
=<API key string here>
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch
query%3Ddc.title%2Bany%2Bdarwin%2BAND%2Bdc
rtKeys%3Dtitle%26httpAccept%3Dapplication/
cked%26api_key=<API key string here>
```

A fragment of the unpacked result:

```
"entry": [
            {
                "title": "Evolution pione
200, Darwin 215",
                "link": "http://dx.doi.or
                "id": "http://dx.doi.org/
                "updated": "2011-07-05T18
                "content": null,
                "dc:identifier": "doi:10.
                "dc:title": "Evolution pi
at 200, Darwin 215",
                "prism:productCode": "nat
                "dc:creator": [
                    "William E. Friedman"
                ],
                ...
            }
...]
```

- Request JSON results, unpacked:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch
queryType=cql&query=dc.title+any+darwin+AN
Keys=title&httpAccept=application/json&rec
key string here>
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch
query%3Ddc.title%2Bany%2Bdarwin%2BAND%2Bdc
rtKeys%3Dtitle%26httpAccept%3Dapplication/
26api_key=<API key string here>
```

The same record packed:

```
"entry": [
            {
                "title": "Evolution pione
200, Darwin 215",
                "link": "http://dx.doi.or
                "id": "http://dx.doi.org/
                "updated": "2011-07-05T18
                "content": null,
                "sru:recordSchema": "info
                "sru:recordPacking": "pac
                "sru:recordData": {
                    "pam:message": {
                        "pam:article": {
                            "xhtml:head":
                                "dc:ident
"doi:10.1038/461167b",
                                "dc:title
celebrating Lamarck at 200, Darwin 215",
                                "prism:pr
                                "dc:creat
                                    "Will
                                ],
        ...
            }
```

```
    ...]
```

Name:	recordSchema
Value:	A URI identifiying the record schema to which results should conform

Notes

SRU-defined parameter. Currently only one record schema is supported, PRISM Aggregator Message or PAM. For more information about PAM, see the PAM 2.1 Guide. Set by default, therefore this parameter is not used.

Name:	sortKeys
Value:	A supported sort specifier, one of those from the list below, specified as indicated

Notes

SRU-defined parameter. Specifies how the results should be ordered. A number of standard specifiers are supported, enabling the caller to request results ordered by appropriate criteria, for example by relevance or by title. Different ordering criteria are encoded in the different specifiers.

Supported criteria and their matching specifiers are:

Ordering criterion	Specifier
By relevance	,pam,0
By newest	publicationDate,pam,0
By oldest	publicationDate,pam,1
By title	title,pam,1
By journal	publicationName,pam

Example

- Order results by title:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch?
queryType=cql&query=dc.title+any+darwin+OR
eys=title,pam,1&api_key=<API key string he
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch?
query%3Ddc.title%2Bany%2Bdarwin%2BOR%2Bdc.
tKeys%3Dtitle%2Cpam%2C1%26api_key=<API key
```

- Order results by publication date, oldest first:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch?
queryType=cql&query=dc.title+any+darwin+OR
eys=publicationDate,pam,1&api_key=<API key
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch?
query%3Ddc.title%2Bany%2Bdarwin%2BOR%2Bdc.
tKeys%3DpublicationDate%2Cpam%2C1%26api_ke
```

Name:	startRecord
Value:	Number, 1 or greater

Notes

SRU-defined parameter. Specifies the number of the first result that should be returned. Used together with maximumRecords, allows callers to paginate the returned results. Note that the value is interpreted as a position in the sequence of all results that match the search query. The first position in the sequence is 1, i.e. the first record; therefore the value supplied MUST be greater than 0.

If this parameter is not supplied, the default value is 1.

Example

- Order results by title, starting from the 5th record:

Complete unencoded request including API key:

```
http://api.nature.com/content/opensearch?
queryType=cql&query=dc.title+any+darwin+OR
eys=title,pam,1&startRecord=5
```

Complete request including API key, encoded:

```
http://api.nature.com/content/opensearch?
query%3Ddc.title%2Bany%2Bdarwin%2BOR%2Bdc.
tKeys%3Dtitle%2Cpam%2C1%26startRecord%3D5
```

Name:	stylesheet
Value:	This parameter is not used

Notes

SRU-defined parameter. This parameter is not used and will be ignored by the server.

Name:	version
Value:	Legacy parameter

Notes

Legacy, defined by the SRU standard. While supported, this parameter is intended only for use by legacy applications.

Unsupported parameters

The following parameters are defined by the SRU standard but are NOT supported:

- recordXPath
- resultSetTTL
- extraRequestData

Nature.com Developers