Digital Commons API Version 2: Real-Time API

The Digital Commons API v2 allows you to retrieve metadata for your repository to use in a variety of applications.

Version 2 of the API is a real-time API continuously updated with the latest changes and revisions in Digital Commons. It provides more flexibility in formulating various queries, including the ability to return subsets of metadata fields.

As a RESTful API, the Digital Commons API uses a base URL and standard HTTP methods such as GET and PUT. Each request requires a special header—a security token—which you can request from your Consultant at dc-support@bepress.com. If you already have a security token from either API v1 or v2, you can continue using it.

Any or all indexed metadata (including hidden metadata) can be returned in API results by creating queries that use Solr-based field search criteria. The results are returned as a single JSON file.

If you have any questions about the real-time API, please contact Consulting Services at dc-support@bepress.com.

Contents

Basics

API base URL

The base URL for making Digital Commons API calls is:

```
https://content-out.bepress.com/v2/{site_url}
```

Endpoints and parameters listed in this document are appended to this URL when making API requests, following this example:

```
https://content-out.bepress.com/v2/{site_url}/query?q=video
```

Four endpoints are available: query, export, download, and fields. Each is described below in more detail.

Request URLs must be called along with the header described under "Authorization" using an HTTP client or your choice of programming language. **Requests will not work if pasted into a browser.**

Authorization

You will need to include a security token as a header with each request. Please contact Consulting Services to obtain your security token. (If you already have a security token from any version of the API, you do not need a new one.)

"Authorization" is the header name for the security token. The header follows this pattern:

Authorization: `{security_token}`

Replace `{security_token}` with the security token you received from your Consultant. Remember to include the security token in the headers of each request and not in the request URL.

Params ●	Authorization	Headers (8)	Body	Pre-request Script	Tests	Settings	
Headers	👁 7 hidden						
	KEY		**VALUE**			**DESCRIPTION**	
☑	Authorization		0123456789abcde				
	Key		Value			Description	

Making an API request

You may use your preferred client or programming language to call the API. This documentation includes instructions for a free client called Postman. Other common clients (also free) include HTTPie and Insomnia.

API requests follow this pattern, with a method (GET in this case), your site URL, your desired endpoint, and an authorization header:

```
GET  https://content-out.bepress.com/v2/{site_url}/{endpoint}
Authorization: {security_token}
```

Here are the basic steps for making an API request:

1. Combine the API base URL with your site URL and desired endpoint. Endpoints are listed in this documentation along with example request URLs.

2. Replace `{site_url}` with the URL of your Digital Commons site minus https://, e.g., your.institution.edu.

3. Include the Authorization in the request headers, as described above.

4. Enter parameters at the end of the request URL as described in the instructions for each endpoint below.

Making requests in Postman

If using the Postman client, here are some pointers for where to put the various components of an API request.

1. Select the HTTP method (GET or PUT) for your request using the Postman method dropdown menu. The endpoints listed in this documentation specify the HTTP method for each type of request.

2. Add the URL request to the Postman request window. Replace "ENDPOINT" with your desired endpoint. Example URL requests are provided for the endpoints listed in this document.

3. Click on Headers to add your security token: enter Authorization as the Key and your security token in the Value field.

4. If creating a search query, add those query parameters to the URL request.

5. Click the "Send" button.

Retrieving JSON files

After making your request, you will receive results in the response window.

To download the results of the API response, click on "Save Response" and select "Save to a file." That will download the results into a JSON file that contains the data you requested.

Create queries

Get current results from your site that meet specific query parameters. Queries return up to 100 results by default. You can adjust this number from 1 to 1000 using the "limit" parameter, described below. To retrieve all results, you may use the "export" endpoint (see the "Export all results meeting your parameters" section below).

You can retrieve "query" results as often as you'd like, and the data is updated in real time as changes and revisions are indexed in the Solr database.

Endpoint

```
/query
```

Example request URL

```
https://content-out.bepress.com/v2/{site_url}/query?q=Video&fields
=title,abstract
```

Method

GET

Make a GET request

1. Make a GET request with the "query" endpoint. Replace `{site_url}` with the URL of your Digital Commons site minus "https://". Include the security token as a header. See "Making an API Request" above for detailed steps.

2. Add query parameters after the "query" endpoint:

    a. **?q** – This parameter allows you to run a free form text query that will return results for all items that contain that value in any metadata field. For example, `https://content-out.bepress.com/v2/{site_url}/query?q=video` would return results that contain "video" in any metadata field.

    b. **?{metadata field}** – If you would like to search specific fields for specific values, this query will return results for items that contain the specified value(s) in the specified field(s).

        - For example, `https://content-out.bepress.com/v2/{site_url}/query?abstract=water` would return results in which "water" was contained in the abstract.

        - An example of a query that returns works by a single author is `https://content-out.bepress.com/v2/{site_url}/query?author="Jeanne S. Authorson"`

        - Example of a query for a publication year: `https://content-out.bepress.com/v2/{site_url}/query?publication_year=2019`

- It is also possible to run queries that return metadata for a specific item, using its context_key as a filter. The context_key for an article can be found when viewing results that contain all available metadata fields (refer to the "select_fields" parameter below). An example of a query that would return a single article is `https://content-out.bepress.com/v2/{site_url}/query?context_key=123`

c. **?parent_link** – Retrieve all items in a publication context like a series or journal, using the full publication URL as a filter. (Note: most URLs in system metadata start "http:" even if they are served using "https:" protocol).

- The format of a query that would return metadata for items in a collection is `https://content-out.bepress.com/v2/{site_url}/query?parent_link=http://{site_url}/{publication}`

- Example for a "faculty_pubs" series: `https://content-out.bepress.com/v2/site.institution.edu/query?parent_link=http://site.institution.edu/faculty_pubs`

d. **?fields** – If you would like to have a specific set of metadata fields returned, use the "fields" parameter. Follow the "fields" parameter with an equal sign and a comma-separated list (no spaces) of the metadata fields you wish to be returned. Be sure to use the back-end label for the metadata field (see the "fields" endpoint to learn how you can get a list of the back-end field names). For example, `https://content-out.bepress.com/v2/{site_url}/query?fields=url,title` should return only the URL and Title fields for each record that has them, but no other fields. If this parameter is omitted, then a default set of fields will be returned.

e. **?select_fields** – If you would like to return all available metadata fields, use select_fields=all. For example, `https://content-out.bepress.com/v2/{site_url}/query?select_fields=all` will return all indexed metadata fields for all items in your repository. Alternatively, if you use select_fields=default, a list of the most commonly used metadata fields will be returned (see information on "fields" endpoint to learn how you can get a list of the default fields). NOTE: **select_fields** and **fields** parameters are mutually exclusive; only one of them should be used at a time.

f. **?limit** – Specifies how many results each page should contain; 100 results are returned by default. The limit can be set from 1 to 1000 results per page. The method will return up to the specified number of results. If available results are less than the specified limit, all results will be returned. For example, `https://content-out.bepress.com/v2/{site_url}/query?limit=500` will return the first 500 results. If you want to export all results at once, use the "export" endpoint (see below for instructions).

g. **?start** - Specifies from which result the page should start; the default value is 0. For example, `https://content-out.bepress.com/v2/{site_url}/query?start=101` will return results starting with the 101st result. If the start parameter is set beyond the actual number of results available, the method will return no items in the results section.

h. **Combining multiple parameters:** You can add multiple query parameters in the URL separated by an ampersand (&) to further refine your results.

For example, this will return the URL, title, abstract, and author for the first 500 results that contain "water" in any metadata field:

```
https://content-out.bepress.com/v2/{site_url}/query?q=water&fields
=url,title,abstract,author&limit= 500
```

This will return all results within the indicated publication that contained "water" in the metadata:

```
https://content-out.bepress.com/v2/{site_url}/query?q=water
&parent_link=http://{site_url}/{publication}
```

3. Save the JSON file that has been generated with your data. See "Retrieving JSON Files" above for basic instructions, plus note the following when using the "query" endpoint: **By default, only the first 100 results will be returned; if you wish a larger number of results to be returned, you can specify that with the "limit" parameter in the URL request.**

4. Open the JSON file using your preferred program.

Sample response

```
1  {
2      "results": [
3          {
4              "title": "Increasing Production of <i>Copepod nauplii</i> in a Brown-Water
                   Zooplankton Culture With Supplemental Feeding and Increased Harvest
                   Levels",
5              "abstract": "<p>Two experiments were performed to investigate increasing the
                   number of copepod nauplii obtained from a brown-water zooplankton
                   culture for feeding to larval red snapper Lutjanus campechanus.
                   Brown-water zooplankton cultures rely on the regular exchange of (brown)
                   estuarine water to nourish and harvest the copepods. The majority of
                   copepods in both experiments were Acartia tonsa. The goal of the first
                   experiment was to determine whether adding rice bran to a brown-water
```

Where "results" lists the results of the query based on which parameters you have specified in the query.

"Query_meta" will appear at the bottom of the results and list information about the query itself, such as the total number of available results (total hits) and the query parameters.

```
                        rlower Garden banks can runction as repositories and sources or reer
                        biota.</p>\n"
389    ·····}
390    ····],
391    ····"query_meta": {
392    ·······"total_hits": 211,
393    ·······"start": 0,
394    ·······"limit": 100,
395    ·······"field_params": {
396    ··········"include_only": [
397    ·············"title",
398    ·············"abstract"
399    ··········]
400    ·······}
401    ····}
402  }
```

Export all results meeting your parameters

The "export" endpoint creates a snapshot of all available records meeting your parameters and generates an export ID. You can then use the export ID to retrieve that snapshot in a JSON file using the "download" endpoint. Results of an export request remain active for 90 days.

"Export" uses the same parameters as the "query" endpoint above (except for the "start" and "limit" parameters, which don't apply since the "export" endpoint always returns all results).

There are two steps to exporting all results that meet your specified parameters into a JSON file.

Export step one: Create the file to be exported

Endpoint

```
/export
```

Example request URL

```
https://content-out.bepress.com/v2/{site_url}/export?q=Video&fields
=title,abstract
```

Method

PUT

Make a PUT request

1. Make a PUT request with the "export" endpoint. Replace `{site_url}` with the URL of your Digital Commons site minus "https://". Include the security token as a header. See "Making an API Request" for detailed steps.

2. Add query parameters after the "export" endpoint; see query parameters above. Just as with the "query" endpoint, you can use multiple parameters to refine your results. As mentioned, because all results are exported, the "start" and "limit" parameters will be ignored.

3. This query will return an "ExportId" result like the one below.

Sample response

```
1  {
2      "ExportId": "1607042567-a9ebf88f-fa08-4f3a-9998-5cbd3ff8a05f"
3  }
```

Export step two: Download the file

Endpoint

```
/download
```

Example request URL

```
https://content-out.bepress.com/v2/{site_url}/download/1607042567-a9ebf88f-
fa08-4f3a-9998-5cbd3ff8a05f
```

Method

GET

Make a GET request

1. Make a GET request with the "download" endpoint. Replace `{site_url}` with the URL of your Digital Commons site minus "https://". Include the security token as a header. See "Making an API Request" for detailed steps.

2. Add the export ID generated in the previous step after the "download" endpoint.

3. This query will return your exported results in a JSON file, which you can then save; see "Retrieving JSON files" above.

Sample response

```
1  {
2  ····"results": [
3  ········{
4  ············"title": "An Ecological Model of the Habitat Mosaic in Estuarine Nursery
              Areas: Part I-Interaction of Dispersal Theory and Habitat Variability
              in Describing Juvenile Fish Distributions",
5  ············"abstract": "<p>Identification of critical habitat in estuarine nursery
              areas is an important conservation and management objective. Habitat
              can be viewed as a mosaic of both temporally variable environmental
              features and spatially variable structural features that combine to
              define optimal habitat. Effective models of juvenile distributions
```

Export full site snapshots

To retrieve all available metadata (including hidden metadata) for all published content in your repository, follow the instructions above for exporting and downloading results of your query into a JSON file (see "Export all results meeting your parameters"). Add the "select_fields=all" query after the "export" endpoint to generate a full site snapshot.

Endpoint

```
/export
```

Example request URL

```
https://content-out.bepress.com/v2/{site_url}/export?select_fields=all
```

Retrieve a list of metadata fields

Request a list of either default metadata fields or all available metadata fields in the repository.

Endpoint

```
/fields
```

Example request URL

```
https://content-out.bepress.com/v2/{site_url}/fields?field_type=all
```

Method

GET

Make a GET request

1. Make a GET request with the "fields" endpoint. Replace `{site_url}` with the URL of your Digital Commons site minus "https://". Include the security token as a header. See "Making an API Request" for detailed steps.

2. Add query parameters after the "fields" endpoint:

   a. **?field_type=all** – Returns a list of all fields used in the repository. Example:

   ```
   https://content-out.bepress.com/v2/{site_url}/fields?field_type=all
   ```

   b. **?field_type=default** – Returns a standard set of default fields. Example:

   ```
   https://content-out.bepress.com/v2/{site_url}/fields?field_type=default
   ```

3. This query will return your results which you can then save; see "Retrieving JSON files" above.

Sample response

```
1   {
2       "fields": [
3           "author",
4           "document_type",
5           "comments",
6           "publication_date",
7           "title",
8           "url",
```

Metadata fields available in API v2

All metadata indexed in Digital Commons is available via the API. This encompasses most metadata fields including custom fields. The following fields currently are not indexed and won't appear in API results.

- Keywords
- Geolocate
- Latitude
- Longitude
- Issn
- Volnum
- Issnum
- Artnum
- Fpage
- Lpage
- Doi
- Submission_date
- Publisher

If you have any questions, please contact our Consulting Services team at dc-support@bepress.com