

Dataverse API

This page outlines some common uses of the Dataverse API within the Scholars Portal Dataverse community with step-by-step instructions to run these commands.

To find out more about the Dataverse API, [see the official documentation](#).

- [Using the API](#)
- [Testing the API](#)
- [Getting Your API Token](#)
- [Viewing JSON in Your Browser](#)
- [Sending Output to a File](#)
- [Super User Commands](#)
- [Finding a Dataset ID](#)
- [Finding a Dataverse ID](#)
- [Some Common & Useful API Commands](#)
 - [Find the date a Dataverse collection was created](#)
 - [Changing a citation date](#)
 - [Get the size of a collection](#)
- [Bulk Uploading](#)
- [Bulk Downloading a Larger Dataset](#)
 - [Download restricted data](#)
 - [Download a folder](#)
 - [Download a version](#)

Using the API

How you use an API can depend on your operating system, permissions on your local machine, and other tools you are familiar with. There are many ways that you can work with API, including using your built-in command line interface, with programming languages like Python, or through applications for working with APIs like [Postman](#).

Most of the examples in the official documentation use a tool called **curl**, which can be used in your machine's command-line environment (shell). To use curl:

- In Windows, you can download and install [Git for Windows](#), which contains a tool called GitBash.
- In Windows 10, you may be able to enable the [Windows Subsystem for Linux](#).
- On Mac/Linux, the default "Terminal" application can be used to run curl commands.

See [the Library Carpentry Unix Shell Setup](#) page for some additional setup steps and tips.

To learn more about working with the command line, see the [Programming Historian "Introduction to the Bash Command Line" lesson](#).

Once you have an environment set up, a quick way to test if it is working correctly is with an API command like the following, which retrieves information about a collection (in this case, the root collection on the demo site):

```
curl https://demodv.scholarsportal.info/api/dataverses/sp
```

After entering this command and pressing enter, you should see JSON-formatted data with details about the collection, like the description and the date it was created.

Testing the API

It's generally a good idea to test out the API commands you're using first, especially if you are new to using APIs or are not sure how a command works. To get started, we suggest you try out your API commands in [the demo environment](#) first, especially if those commands can't be reversed (e.g. deleting content). An API command run on the command line will not usually present you with an option to "confirm" your action, so we **strongly recommend** testing it first!

Getting Your API Token

To complete many tasks with the Dataverse API, you will need to have your account's API token. To retrieve your API token in Dataverse:

1. Log into Dataverse
2. Click your username in the upper right corner of Dataverse
3. In the list that opens, click "API Token"
4. Copy the API Token you see on this page, or click "Create Token" if one has not already been generated.

API tokens should always be stored in a secure way, like a password, as they give complete access to all of the data in your account.

Viewing JSON in Your Browser

Many API commands can involve viewing JSON-formatted information directly in your web browser. While some browsers will format this in a readable way by default (e.g. Firefox), some will not (e.g. Chrome). If you find the JSON hard to read, consider installing a JSON viewing browser extension (e.g. [JSON Formatter for Chrome](#)).

Sending Output to a File

If a command you are using outputs data that you want to save, you can save the results to a file instead of having them displayed to you on the command-line. To do this, you need to adjust your command to send the output to a file. The general structure to do this is:

```
curl https://some/api/command > filename.extension
```

For example, if you wanted to send the JSON output of information about a collection to a file, you would do the following:

```
curl https://dataverse.scholarsportal.info/api/dataverses/ottawa > ottawa.json
```

You could adjust the file extension or filename by changing the "ottawa.json" portion of the command.

Super User Commands

Some API commands in Dataverse can only be run by a super user. This means that this command has to be run by someone from Scholars Portal. If you have a command like this that you want to run, contact us.

Finding a Dataset ID

For API commands related to datasets, you may need to get the ID of the dataset, which is different from the dataset's persistent identifier. To find the ID of a dataset:

1. Find the permanent identifier (DOI or Handle) for your dataset
2. Use the following API command directly in your browser, replacing the DATAVERSEURL and DATASETIDENTIFIER with your own: <https://DATAVERSEURL/api/datasets/:persistentId/?persistentId=DATASETIDENTIFIER>
 - a. For example, for this dataset, we would use the following: <https://dataverse.scholarsportal.info/api/datasets/:persistentId/?persistentId=doi:10.23685/1H9TOV>
3. In the JSON that is shown, we see the data ID for this dataset is 97164. The dataset ID is the first ID in the JSON output.

Finding a Dataverse ID

For API commands related to collections, you may need the ID of the collection. This is the identifier that is in the URL of your collection.

For example, for the UBC Dataverse collection, which is available from <https://dataverse.scholarsportal.info/dataverse/ubc>, the ID of the collection would be "ubc".

Some Common & Useful API Commands

Find the date a Dataverse collection was created

To get more details about a specific Dataverse collection, such as the date it was created, you can enter the following API call as a URL in a web browser, replacing the \$ALIAS in the URL with the alias of the Dataverse collection:

```
https://dataverse.scholarsportal.info/api/dataverses/$ALIAS
```

For example, if we wanted to find more information about the "ottawa" Dataverse collection, we would use the following URL:

```
https://dataverse.scholarsportal.info/api/dataverses/ottawa
```

If the Dataverse collection has not been published, you will need to add an API token to the URL:

```
https://dataverse.scholarsportal.info/api/dataverses/$ALIAS?key=$API_TOKEN
```

Changing a citation date

1. Make note of the DOI or handle of the dataset you want to change the citation date for (referred to as the "persistent ID" below).
2. Ensure that the date you want to change the citation date to is reflected in your dataset metadata in a date field, e.g. in the deposit date field, and is in a published version of the dataset.
3. Find the name of the field you want to use as the citation date in the Metadata tab under "Export Metadata" > "JSON", e.g. for the deposit date you would use the field name of 'dateOfDeposit', for production date, use "productionDate".
 - a. <https://demodv.scholarsportal.info/api/datasets/:persistentId/?persistentId=doi:10.5072/FK2/QLQNAP> (In this example, we can see that the field is 'dateOfDeposit' under the metadata blocks.

4. Enter the following command, replacing the persistent ID (starting with "doi:" or "hdl:"), API key, and name of the metadata field being used to replace the citation date:

```
curl -H "X-Dataverse-key: API_TOKEN" -X PUT "https://dataverse.scholarsportal.info/api/datasets/:persistentId/citationdate?persistentId=$PERSISTENT_ID" --data "dateOfDeposit"
```

Get the size of a collection

To get the size of a collection which you have admin access to, paste the following command in your web browser, replacing the API Token and the Dataverse ID (found in the URL of a collection):

https://dataverse.scholarsportal.info/api/dataverses/DATAVERSE_ID/storagesize?key=API_KEY

To get this through the command line, use the following command:

```
curl -H "X-Dataverse-key: API_TOKEN" https://dataverse.scholarsportal.info/api/dataverses/DATAVERSE_ALIAS/storagesize
```

This API call will return a total size in bytes. You can convert it to MB/GB using an online tool like this one (note that the built-in Google unit converter uses an inaccurate formula for this): <https://www.convertunits.com/from/byte/to/gigabyte>

Bulk Uploading

When used in a script, API commands can be combined or used within loop structures to bulk upload data, such as multiple files within a folder. One example of this can be seen in [this script](#), which can be used to create multiple datasets at a time within a collection.

Bulk Downloading a Larger Dataset

If you are downloading a dataset which has 5GB of data or more, you will not be able to download all of the files at once through the user interface. One way around this is to use a tool called `wget` in your command-line in order to download all of the files in the dataset to a local folder on your machine. In some Windows environments, you may need to install `wget` before being able to use this command.

*Please note the file hierarchy is not maintained in the downloaded dataset.

First, you will want to create a new folder for the dataset you're downloading, and navigate to this directory in your terminal.

If the dataset has any restricted files, you will need to retrieve your API token in order to download them (see more details below). If no API token is used, then this command will only download the files you have access to.

You will also need the identifier (DOI or handle) for the dataset you want to download. This should be formatted as either (for example) "doi:10.5683/SP3/OHVUDH" or "hdl:10864/10120".

Depending on how large the dataset is, this may take a while to run. [More information about this command is available in the documentation.](#)

To download all of the files in a dataset, use the following command, replacing the IDENTIFIER with your own:

```
wget -r -e robots=off -nH --cut-dirs=3 --content-disposition "https://dataverse.scholarsportal.info/api/datasets/:persistentId/dirindex?persistentId=IDENTIFIER"
```

Download restricted data

If the dataset has any restricted files, you will need to be given access to the files and retrieve your API token in order to download them. To add your API key to the command, use the following formatting, replacing the IDENTIFIER and API-KEY with your own:

```
wget -r -e robots=off -nH --cut-dirs=3 --header "X-Dataverse-key: API-KEY" --content-disposition "https://dataverse.scholarsportal.info/api/datasets/:persistentId/dirindex?persistentId=IDENTIFIER"
```

Download a folder

If the dataset has a file structure in place, this command can be adjusted to download a specific folder. To download a folder, add the "folder" parameter to the URL and replace the IDENTIFIER and FOLDER-NAME with your own:

```
wget -r -e robots=off -nH --cut-dirs=3 --content-disposition "https://dataverse.scholarsportal.info/api/datasets/:persistentId/dirindex?persistentId=IDENTIFIER&folder=FOLDER-NAME"
```

If the folders are nested, use the complete path to the folder, e.g. "FOLDER1/FOLDER2/FOLDER3".

Download a version

If the dataset has more than one version, and you don't want to download the latest published version, you can add the version to your URL, such as "1.0". To download a folder, add the "folder" parameter to the URL and replace the IDENTIFIER and VERSION-NUM with your own:

```
wget -r -e robots=off -nH --cut-dirs=3 --content-disposition "https://dataverse.scholarsportal.info/api/datasets/:persistentId/dirindex?persistentId=IDENTIFIER&version=VERSION-NUM"
```