

SEADUploader: Using SEAD's API for bulk uploads and application integration

SEAD's Clowder component has a service api that allows you to read/write/annotate/tag/delete datasets and collections (among other things). These services are used within SEAD's bulk upload/sync command line tool and can be called from Curl or your own code to allow your application to read/write directly to/from Clowder spaces.

SEAD Uploader

The SEAD Uploader is a command line tool that can be used to upload or sync portions of your disk with a Clowder project space. (The upload speed you achieve will depend on your disk and network speed, but the client itself can manage >100K file uploads and we've seen at least 250MB/minute transfer speeds over a mix of large and small files.).

Steps to get started:

1. Please make sure you have JAVA installed on your computer. If you don't, you can download it here: <https://java.com/en/download/>
2. Download the SEAD Uploader jar file : [sead2.3.jar](#).
3. On your computer, open the root directory that contains files and folders you would like to upload to your Project Space in SEAD.
4. Put the sead2.3.jar file in this directory.
5. Holding the SHIFT button, right click on your mouse to open a menu. Select the "Open Command Window Here" option.
6. Once you have the Command Window open, invoke the SEADUploader by typing the following in the Command Window:

```
java -cp sead2.3.jar org.sead.uploader.clowder.SEADUploader <-listonly> <-limit=<X>> <-skip=<n>> <-verify> <-ex=<Y>> <-key=<apiKey>> <-fornew> -server=<serverUrl> -id=<id> <directories list...>
```

where:

-listonly: write information about what would/would not be transferred without doing any upload

-limit=<X>: limit this run to at most X dataset uploads (any required collections will be automatically created)

-skip=<n>: skip the first <n> files found on disk before starting to check whether files exist on the server and uploading those that are not yet in SEAD (any required collections will be automatically created)

-ex=<Y>: exclude any file that matches the provided regular expression pattern, e.g. -ex=^\. (exclude files that start with a period) -ex*.txt (exclude all files ending in .txt). Multiple repeats of this flag can be used to exclude based on multiple patterns.

-fornew: By default, the Uploader will search for an existing dataset with the same name as the specified directory and only upload files that are not already in that dataset. -fornew will always create a new dataset and, subject to other settings, will therefore always upload all files.

-key=<apiKey>: use an API Key created you create (on your Profile page) to avoid having to enter a username/password.

-id=<id>: if you know a dataset exists, specifying it's id here will improve performance as the uploader won't have to scan through all datasets to find it.

-server=<serverUrl>: the base URL of the SEAD/Clowder server you're interacting with, e.g. -server=https://sead2.ncsa.illinois.edu

directories list - a list of one or more directory names the Uploader should work on. The Uploader will recurse (depth first) through the files and subdirectories contained within any listed directory.

-verify: adding this flag will cause the uploader to use a cryptographic hash to verify that the local file and the copy in SEAD are exactly the same, byte-for-byte. With SEAD 2.0, this flag should be used on a second run of the Uploader - hashes in 2.0 are calculated asynchronously and may not be available immediately after a file is uploaded.

SEAD recommends using an API key rather than your username/password.

Examples:

2.0 examples: use of the Uploader creates one Dataset with Folders and Files inside

```
java -cp sead2.3.jar org.sead.uploader.clowder.SEADUploader -listonly -server=https://sead2.ncsa.illinois.edu mydir
```

Using SEAD's 2.0 instance, check ./mydir and list the Dataset and all Folders and Files that would be created without the -listonly flag

```
java -cp sead2.3.jar org.sead.uploader.clowder.SEADUploader -server=https://sead2.ncsa.illinois.edu mydir
```

Using SEAD's 2.0 instance, create a 'mydir' Dataset in your account on sead2.ncsa.illinois.edu and create Folders and Files for all contained items. Each File will be annotated with metadata indicating the original path (user metadata: "instanceOf (<http://purl.org/vocab/frbr/core#embodimentOf>) with the value /mydir for the mydir directory in this example and /mydir/<relative path> for all Files). Note that if this command is run without the -merge flag, it will create a new Dataset and upload all folders and files again, even if a Dataset already exists from a prior run (which is not usually what you'd want). -merge makes the Uploader check for an existing Dataset and only uploads missing content, or folders/files added to your local directory since the last upload.

```
java -cp sead2.3.jar org.sead.uploader.clowder.SEADUploader -limit100 -server=https://sead2.ncsa.illinois.edu mydir
```

Using SEAD's 2.0 instance, locate/create a 'mydir' Dataset in your account on sead2.ncsa.illinois.edu and create Folders and Files for the first 100 contained items. Running the command again will add the next 100 items (because of the -merge flag it will find/skip the ones already uploaded).

```
java -cp sead2.3.jar org.sead.uploader.clowder.SEADUploader -verify -server=https://sead2.ncsa.illinois.edu mydir
```

Using SEAD's 2.0 instance, verify that all files previously uploaded to SEAD 2 are exactly the same as those on your disk. Since SEAD 2 currently calculates hash values with a delay, the -verify flag should be added on a second run of the Uploader.

SEAD 2.0 Authentication Process

Uploading data to SEAD 2.0 requires an APIKey or a local SEAD username/password (social logins are not supported). In username/password mode, each time you run the Uploader, you will be prompted for your username and password. Your password will be transmitted via https to the SEAD2 server, but no copy is stored on your local machine.

Help

If you experience problems with using the SEADUploader, please contact SEAD at SEADdatanet@umich.edu and we will be happy to walk you through the steps.