ldcoolp-figshare

Release v0.3.2

Chun Ly, UA Research Data Repository (ReDATA) Team

Jul 07, 2021

CONTENTS:

OVERVIEW

`ldcoolp-figshare` is a command-line API used in LD-Cool-P to interact with the Figshare API. It is used by ReDATA data curators for the curation of submitted research data products. This codebase is generalized such that other curators with a Figshare for Institution instance can use.

1.1 Getting Started

You will need Python. We recommend v3.9. This code will be tested with v3.7, v3.8, and v3.9.

This code is published as `ldcoolp-figshare` on PyPI. Thus, the simplest installation is:

```
pip install ldcoolp-figshare
```

1.1.1 Requirements

All requirements will be given in the requirements.txt

1.2 Use Examples

The primary class/object that is used for all interactions with the Figshare API is *FigshareInstituteAdmin*. Below are for how to use this API:

```python
from ldcoolp_figshare import FigshareInstituteAdmin

token = "***ENTER YOUR API KEY ***"
stage = False  # Set to False (True) for production (stage) instance

fs_admin = FigshareInstituteAdmin(token=token, stage=stage)
```

There are several methods available with `FigshareInstituteAdmin`. We refer users to the full API documentation for more details. Below we provide some examples to get users started.

1.2.1 Get a list of accounts

To retrieve a list of accounts for an institution:

```
fs_admin.get_account_list()
```

Note that this provides you with the `account_id` of a user

1.2.2 Obtain curation records

To retrieve a full list of curation records (of any state) for an institution:

```
curation_df = fs_admin.get_curation_list()
```

If you wish to retrieve all curation records for a specific item/deposit, then provide the `article_id`:

```
article_id = 12345678
article_curation_df = fs_admin.get_curation_list(article_id)
```

To obtain more information about a specific curation record:

```
curation_id = 1234567
details_dict = fs_admin.get_curation_details(curation_id)
```

1.2.3 DOI status and reservation

To check if a DOI is reserved for an item/deposit:

```
article_id = 12345678
check, DOI_string = fs_admin.doi_check(article_id)
```

Here, `check` will either be `False` (no DOI) or `True`.

Alternatively, you can reserve a DOI if it hasn't been done so. This will provide an prompt before performing the task.
Note: This step is irreversible!

```
article_id = 12345678
check, DOI_string = fs_admin.reserve_doi(article_id)
```

1.2.4 Retrieve list of institution groups

```
group_df = fs_admin.get_groups()
```

1.2.5 Retrieve list of articles for a user

```
account_id = 98765432
article_df = fs_admin.get_user_articles(account_id)
```

1.3 Continuous Integration

Under construction.

1.4 Versioning

We use SemVer for versioning. For the versions available, see the tags.

Releases will be auto-generated using this GitHub Actions script following a `git tag vX.Y.Z`.

1.5 Authors

- Chun Ly, Ph.D. (@astrochun) - University of Arizona Libraries, Research Engagement

See also the list of contributors who participated in this project.

1.6 License

This project is licensed under the MIT License - see the LICENSE file for details.

1.7 API Documentation

1.7.1 ldcoolp-figshare package

The `FigshareInstituteAdmin` Class

class FigshareInstituteAdmin(*token*, *stage=False*, *admin_filter=None*, *log=<Logger stdout_logger (INFO)>*)
Bases: `object`

A Python interface for administration and data curation with institutional Figshare instances

Most methods take an `article_id` or `curation_id` input

> **Parameters**
>
> - **token** (`str`) – Figshare OAuth2 authentication token
> - **stage** (`bool`) – Flag to either use Figshare stage or production API. Default: production
> - **admin_filter** (`Optional[list]`) – List of filters to remove admin accounts from user list
> - **log** (`Logger`) – Logger object for stdout and file logging. Default: stdout
>
> **Variables**

- **token** – Figshare OAuth2 authentication token
- **stage** – Flag to either use Figshare stage or prod API
- **baseurl** – Base URL of Figshare API
- **baseurl_institute** – Base URL of Figshare API for institutions
- **headers** – HTTP header information
- **admin_filter** – List of filters to remove admin accounts from user list
- **ignore_admin** – Flags whether to remove admin accounts from user list

doi_check(*article_id*, *process=True*)

Check if DOI is present/reserved for `article_id`.

Uses: https://docs.figshare.com/#private_article_details

Parameters

- **article_id** (`int`) – Figshare article ID
- **process** (`bool`) – Returns JSON content from `redata_request`, otherwise the full request is provided. Default: True

Return type Union[Tuple[bool, str], Response]

Returns Flag to indicate whether DOI is reserved and DOI (empty string if not). Returns the full `requests.Response` if `process=False`

endpoint(*link*, *institute=True*)

Concatenate the endpoint to the baseurl for `requests`

Parameters

- **link** (`str`) – API endpoint to append to baseurl
- **institute** (`bool`) – Flag to use regular of institute baseurl

Return type `str`

Returns URL for HTTPS API

get_account_details(*flag=True*)

Retrieve account details. This includes group association, number of articles, projects, and collections, and administrative and reviewer role flags

Parameters **flag** (`bool`) – Populate administrative and reviewer roles to database

Return type `DataFrame`

Returns Relational database of details of all accounts for an institution

get_account_group_roles(*account_id*, *process=True*)

Retrieve group roles for a given account, `account_id`.

See: https://docs.figshare.com/#private_institution_account_group_roles

Parameters

- **account_id** (`int`) – Figshare *institute* account ID
- **process** (`bool`) – Returns JSON content from `redata_request`, otherwise the full request is provided. Default: True

Return type Union[dict, Response]

Returns Python dictionary of all group roles for a user or the full `requests.Response`

get_account_list(*process=True*)

> Return pandas DataFrame of user accounts.
>
> See: https://docs.figshare.com/#private_institution_accounts_list
>
>> **Parameters process** (bool) – Returns JSON content from `redata_request`, otherwise the full request is provided. Default: True
>>
>> **Return type** Union[DataFrame, Response]
>>
>> **Returns** Relational database of all user accounts for an institution or the full `requests.Response`

get_articles(*process=True*)

> Retrieve information about all articles within institutional instance
>
> See: https://docs.figshare.com/#private_institution_articles
>
>> **Parameters process** (bool) – Returns JSON content from `redata_request`, otherwise the full request is provided. Default: True
>>
>> **Return type** Union[DataFrame, Response]
>>
>> **Returns** Relational database of all articles for an institution or the full `requests.Response`

get_curation_comments(*curation_id*, *process=True*)

> Retrieve comments about specified curation, `curation_id`.
>
> See: https://docs.figshare.com/#account_institution_curation_comments
>
>> **Parameters**
>>
>> - **curation_id** (int) – Figshare curation ID
>>
>> - **process** (bool) – Returns JSON content from `redata_request`, otherwise the full request is provided. Default: True
>>
>> **Return type** Union[dict, Response]
>>
>> **Returns** Python dictionary with curation comments or the full `requests.Response`

get_curation_details(*curation_id*, *process=True*)

> Retrieve details about a specified curation, `curation_id`.
>
> See: https://docs.figshare.com/#account_institution_curation
>
>> **Parameters**
>>
>> - **curation_id** (int) – Figshare curation ID
>>
>> - **process** (bool) – Returns JSON content from `redata_request`, otherwise the full request is provided. Default: True
>>
>> **Return type** Union[dict, Response]
>>
>> **Returns** Python dictionary with curation metadata or the full `requests.Response`

get_curation_list(*article_id=None*, *status=''*, *process=True*)

> Retrieve list of curation records for `article_id`. If not specified, all curation records are retrieved.
>
> See: https://docs.figshare.com/#account_institution_curations
>
>> **Parameters**
>>
>> - **article_id** (Optional[int]) – Figshare article ID
>>
>> - **status** (Optional[str]) – Filter by status of review. Options are: ['', 'pending', 'approved', 'rejected', 'closed']

- **process** (bool) – Returns JSON content from `redata_request`, otherwise the full request is provided. Default: True

> **Return type** Union[DataFrame, Response]

> **Returns** Relational database of all curation records or the full `requests.Response`

get_groups(*process=True*)

> Retrieve information about groups within institutional instance.

> See: https://docs.figshare.com/#private_institution_groups_list

>> **Parameters** **process** (bool) – Returns JSON content from `redata_request`, otherwise the full request is provided. Default: True

>> **Return type** Union[DataFrame, Response]

>> **Returns** Relational database of all Figshare groups for an institution or the full `requests.Response`

get_other_account_details(*account_id*)

> Retrieve ORCID and Figshare account information (among other metadata)

> See: https://docs.figshare.com/#private_account_institution_user

>> **Parameters** **account_id** (int) – Figshare *institute* account ID

>> **Return type** dict

>> **Returns** Dictionary with full account details

get_user_articles(*account_id*, *process=True*)

> Impersonate a user, `account_id`, to retrieve articles associated with the user.

> See: https://docs.figshare.com/#private_articles_list

>> **Parameters**
>>
>> - **account_id** (int) – Figshare *institute* account ID
>>
>> - **process** (bool) – Returns JSON content from `redata_request`, otherwise the full request is provided. Default: True

>> **Return type** Union[DataFrame, Response]

>> **Returns** Relational database of all articles owned by user or the full `requests.Response`

get_user_collections(*account_id*, *process=True*)

> Impersonate a user, `account_id`, to retrieve collections associated with the user.

> See: https://docs.figshare.com/#private_collections_list

>> **Parameters**
>>
>> - **account_id** (int) – Figshare *institute* account ID
>>
>> - **process** (bool) – Returns JSON content from `redata_request`, otherwise the full request is provided. Default: True

>> **Return type** Union[DataFrame, Response]

>> **Returns** Relational database of all collections owned by user or the full `requests.Response`

get_user_projects(*account_id*, *process=True*)

> Impersonate a user, `account_id`, to retrieve projects associated with the user.

> See: https://docs.figshare.com/#private_projects_list

Parameters

- **account_id** (int) – Figshare *institute* account ID
- **process** (bool) – Returns JSON content from `redata_request`, otherwise the full request is provided. Default: True

Return type Union[DataFrame, `Response`]

Returns Relational database of all projects owned by user or the full `requests.Response`

reserve_doi(*article_id*)
Reserve DOI if one has not been reserved for `article_id`.

See: https://docs.figshare.com/#private_article_reserve_doi

Parameters **article_id** (int) – Figshare article ID

Return type `str`

Returns DOI string

INDICES AND TABLES

- genindex
- search