
pycyapi
Release 0.0.14

Computational Plant Science Lab

Dec 31, 2022

INTRODUCTION

1	Installation	1
2	Authenticating	3
3	Commands	5
3.1	Version	5
3.2	Token	6
3.3	User	6
3.4	List	6
3.5	Stat	6
3.6	Pull	6
3.7	Push	7
3.8	Exists	7
3.9	Create	7
3.10	Share	7
3.11	Unshare	8
3.12	Tag	8
3.13	Tags	8
4	Indices and tables	9

INSTALLATION

This project is [available on the Python Package Index](#). To install with pip:

```
pip install pycyapi
```


AUTHENTICATING

The `pycyapi` CLI must obtain an access token to authenticate with CyVerse. The token may be provided to commands via the `--token` parameter, or set as an environment variable `CYVERSE_TOKEN`. An access token can be obtained from the Terrain API by sending a request with basic auth headers (valid CyVerse username and password):

```
GET https://de.cyverse.org/terrain/token/cas
```

A `token` command is provided as convenient alternative to manually obtaining a token:

```
pycyapi token --username <username> --password <password>
```

The token is printed and can be used with the `--token` option to authenticate subsequent commands.

Note: the `token` command is the only one to return plain text — all other commands return JSON.

COMMANDS

To show available commands help run `pycyapi --help`. The following commands are available:

- **version**: Show the current `pycyapi` version.
- **token**: Retrieve a CyVerse authentication token.
- **user**: Retrieve the user's profile information.
- **list**: List files in a collection.
- **stat**: Get information about a file or collection.
- **pull**: Download one or more files from a collection.
- **push**: Upload one or more files to a collection.
- **exists**: Check if a path exists in the data store.
- **create**: Create a collection in the data store.
- **share**: Share a file or collection with another user.
- **unshare**: Revoke another user's access to your file or collection.
- **tag**: Set metadata for a given file or collection.
- **tags**: Get metadata for a given file or collection.

To show usage information for a specific command, run `pycyapi <command> --help`.

3.1 Version

To show the current version of `pycyapi`, use the `version` command:

```
pycyapi version
```

3.2 Token

To request a CyVerse CAS authentication token, use the `token` command:

```
pycyapi token --username <your CyVerse username> --password <your CyVerse password>
```

The token can be passed via `--token (-t)` argument to authenticate subsequent commands.

3.3 User

The `user` command can be used to retrieve public profile information for CyVerse users. For instance:

```
pycyapi user -t <token> wbonelli
```

3.4 List

To list the contents of a collection in the data store, use the `list` command. For instance:

```
pycyapi list -t <token> /iplant/home/shared/iplantcollaborative/testing_tools/
```

3.5 Stat

To view metadata for a particular collection or object in the data store, use the `stat` command. For instance:

```
pycyapi stat -t <token> /iplant/home/shared/iplantcollaborative/testing_tools/
```

3.6 Pull

To download a single file from the data store to the current working directory, simply provide its full path:

```
pycyapi pull -t <token> /iplant/home/shared/iplantcollaborative/testing_tools/cowsay/  
↪ cowsay.txt
```

To download all files from the `/iplant/home/shared/iplantcollaborative/testing_tools/cowsay/` collection to the current working directory, just provide the collection path instead:

```
pycyapi pull -t <token> /iplant/home/shared/iplantcollaborative/testing_tools/cowsay/
```

Optional arguments are:

- `--local_path (-p)`: Local path to download files to
- `--include_pattern (-ip)`: File patterns to include (0+)
- `--force (-f)`: Whether to overwrite already-existing files

3.7 Push

To upload all files in the current working directory to the `/iplant/home/<my>/<directory>` in the CyVerse Data Store, use:

```
picyapi push -t <token> /iplant/home/<username>/<collection>/
```

Optional arguments include:

- `--local_path (-p)`: Local path to download files to
- `--include_pattern (-ip)`: File patterns to include (0+)
- `--include_name (-in)`: File names to include (0+)
- `--exclude_pattern (-ep)`: File patterns to exclude (0+)
- `--exclude_name (-en)`: File names to exclude (0+)

To upload a single file to the data store, provide the `--local_path (-p)` argument. For instance:

```
picyapi push -t <token> /iplant/home/<username>/<collection> -p /my/local/file.txt
```

If only `include_...`s are provided, only the file patterns and names specified will be included. If only `exclude_...`s are present, all files except the patterns and names specified will be included. If you provide both `include_...` and `exclude_...` sections, the `include_...` rules will first be applied to generate a subset of files, which will then be filtered by the `exclude_...` rules.

3.8 Exists

To determine whether a particular path exists in the data store, use the `exists` command. For instance, to check if a collection exists:

```
picyapi exists -t <token> /iplant/home/<username>/<collection>
```

The `--type` option can be provided with value `dir` or `file` to verify that the given path is of the specified type.

3.9 Create

To create a new collection, use the `create` command:

```
picyapi create -t <token> /iplant/home/<username>/<new collection name>
```

3.10 Share

To share a file or collection with another user, use the `share` command:

```
picyapi share -t <token> /iplant/home/<username>/<collection> --username <user to share_↵
↵with> --permission <'read' or 'write'>
```

Note that you must provide both the `--username` and `--permission` flags.

3.11 Unshare

To revoke another user's access to your file or collection, use the `unshare` command:

```
pycyapi unshare -t <token> /iplant/home/<username>/<collection> --username <username>
```

This applies to both read and write permissions for the specified user.

3.12 Tag

To set metadata for a given file object or collection in your data store, use the `tag` command:

```
pycyapi tag <data object ID> -t <token> -a k1=v1 -a k2=v2
```

This applies the two given attributes to the data object (attributes must be formatted `key=value`).

Warning: this command is an overwrite, not an append. We do not support appending tags as there is no Terrain endpoint to add/remove individual metadata attributes. Note also that by default, key/value pairs are passed on the `avus` attribute of the request body rather than `irods-avus`, e.g.:

```
POST https://de.cyverse.org/terrain/secured/filesystem/<ID>/metadata
{
  "irods-avus": [],
  "avus": [
    {
      "attr": "some key"
      "value": "some value",
      "unit": ""
    }
  ]
}
```

To configure `irods-avus` attributes as well as or in place of standard attributes, use the `--irods_attribute` (`-ia`) option. Both standard and iRODS attributes can be used in the same invocation.

3.13 Tags

To retrieve the metadata describing a particular file object or collection, use the `tags` command:

```
pycyapi tags <data object ID> -t <token>
```

This will retrieve standard attributes by default. To retrieve iRODS attributes instead, use the `--irods` (`-i`) option.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`