nexus3-cli Documentation

Thiago Figueiró

Dec 23, 2018

Contents:

CHAPTER 1

Quick Start Guide

CHAPTER 2

Command-line Interface

CHAPTER 3

API

Relevant Nexus 3 API documentation:

- REST API
- Script API

3.1 NexusClient

exception nexuscli.nexus_client.**DownloadError**
Bases: Exception

class nexuscli.nexus_client.**NexusClient**(*url=None*, *user=None*, *password=None*, *config_path=None*)
Bases: object

Interact with Nexus 3's API.

Parameters

- **url** (*str*) – URL to Nexus 3 OSS service.
- **user** (*str*) – login for Nexus service at given url.
- **password** (*str*) – password for given login.
- **config_path** (*str*) – local file containing configuration above in JSON format with these keys: nexus_url, nexus_user and nexus_pass.

base_url
str – as per url argument.

config_path
str – as per arguments.

CONFIG_PATH = '/home/docs/.nexus-cli'

DEFAULT_PASS = 'admin123'

DEFAULT_URL = 'http://localhost:8081'

DEFAULT_USER = 'admin'

delete(*repository_path*, ***kwargs*)
Delete artefacts, recursively if repository_path is a directory.

Parameters

- **repository_path** – location on the repository service.
- **kwargs** – implementation-specific arguments.

Returns number of deleted files. Negative number for errors.

Return type int

download(*source*, *destination*, ***kwargs*)
Process a download. The source must be a valid Nexus 3 repository path, including the repository name as the first component of the path.

The destination must be a local file name or directory.

If a file name is given as destination, the asset may be renamed. The final destination will depend on self.flatten: when True, the remote path isn't reproduced locally.

Parameters

- **source** – location of artefact or directory on the repository service.
- **destination** – path to the local file or directory.
- **flatten** – when True, the remote path isn't reproduced locally.
- **nocache** – Force download of a directory or artefact even if local copy is available and is up-to-date with the version available on Nexus.

Returns number of downloaded files.

download_file(*download_url*, *destination*)
Download an asset from Nexus artefact repository to local file system.

Parameters

- **download_url** – fully-qualified URL to asset being downloaded.
- **destination** – file or directory location to save downloaded asset. Must be an existing directory; any exiting file in this location will be overwritten.

Returns

list(*repository_path*)
List all the artefacts, recursively, in a given repository_path.

Parameters

- **repository_path** – location on the repository service.
- **kwargs** – implementation-specific arguments.

Returns list of artefacts

Return type list

list_raw(*repository_path*)
As per list but returns a generator of raw Nexus artefact objects

read_config()

repositories
RepositoryCollection – instance

rest_url

scripts

set_config(*user*, *password*, *base_url*)

split_component_path(*component_path*)
Splits a given component path into repository, directory, filename.

A Nexus component path for a raw directory must have this format:

```
repository_name/directory[(/subdir1)...][/|filename]
```

A path ending in / means it represents a directory; otherwise it represents a filename.

```
>>> dst0 = 'myrepo0/dir/'
>>> dst1 = 'myrepo1/dir/subdir/'
>>> dst2 = 'myrepo2/dir/subdir/file'
>>> dst3 = 'myrepo3/dir/subdir/etc/file.ext'
>>> split_component_path(dst0)
>>> ('myrepo0', 'dir', None)
>>> split_component_path(dst1)
>>> ('myrepo1', 'dir/subdir', None)
>>> split_component_path(dst2)
>>> ('myrepo2', 'dir/subdir', 'file')
>>> split_component_path(dst3)
>>> ('myrepo3', 'dir/subdir/etc', 'file.ext')
```

Parameters **component_path** (*str*) – the Nexus component path, as described above.

Returns tuple of (repository_name, directory, filename). If the given component_path doesn't represent a file, filename is set to None.

Return type tuple

upload(*source*, *destination*, ***kwargs*)
Process an upload. The source must be either a local file name or directory. The flatten and recurse options are honoured for directory uploads.

The destination must be a valid Nexus 3 repository path, including the repository name as the first component of the path.

Parameters

- **source** – location of file or directory to be uploaded.
- **destination** – destination path in Nexus, including repository name and, if required, directory name (e.g. raw repos require a directory).
- **recurse** – do not process sub directories for uploads to remote
- **flatten** – Flatten directory structure by not reproducing local directory structure remotely

Returns number of files uploaded.

upload_directory(*src_dir*, *dst_repo*, *dst_dir*, ***kwargs*)
Uploads all files in a directory, honouring options flatten and recurse.

Parameters

- **src_dir** – path to local directory to be uploaded
- **dst_repo** – destination repository
- **dst_dir** – destination directory in dst_repo

Returns number of files uploaded

Return type int

upload_file(*src_file*, *dst_repo*, *dst_dir*, *dst_file=None*)
Uploads a singe file to a Nexus repository under the directory and file name specified. If the destination file name isn't given, the source file name is used.

Parameters

- **src_file** – path to the local file to be uploaded.
- **dst_repo** – name of the Nexus repository.
- **dst_dir** – directory under dst_repo to place file in.
- **dst_file** – destination file name.

write_config()

3.2 Exception

exception nexuscli.exception.**NexusClientAPIError**
Bases: Exception

exception nexuscli.exception.**NexusClientConfigurationNotFound**
Bases: Exception

exception nexuscli.exception.**NexusClientCreateRepositoryError**
Bases: Exception

exception nexuscli.exception.**NexusClientDownloadError**
Bases: Exception

exception nexuscli.exception.**NexusClientInvalidCredentials**
Bases: Exception

exception nexuscli.exception.**NexusClientInvalidRepository**
Bases: Exception

exception nexuscli.exception.**NexusClientInvalidRepositoryPath**
Bases: Exception

3.3 Repository

class nexuscli.repository.model.**Repository**(*repo_type*, ***kwargs*)
Bases: object

Creates an object representing a Nexus repository with the given format, type and attributes.

Parameters

- **name** (*str*) – name of the new repository.

- **format** (*str*) – format (recipe) of the new repository. Must be one of data:*nexuscli.repository.validations.KNOWN_FORMATS*.
- **blob_store_name** (*str*) –
- **depth** (*int*) – only valid when `repo_format=yum`. The repodata depth.
- **remote_url** (*str*) –
- **strict_content_type_validation** (*bool*) –
- **version_policy** (*str*) –
- **write_policy** (*str*) – One of :py:data:
- **layout_policy** (*str*) – One of
- **ignore_extra_kwargs** (*bool*) – if True, raise an exception for unnecessary/extra/invalid kwargs.
- **repo_type** – type for the new repository. Must be one of `nexuscli.repository.validations.KNOWN_TYPES`.
- **kwargs** – attributes for the new repository.

Returns the created repository

Return type *Repository*

configuration

class nexuscli.repository.model.**RepositoryCollection**(*client=None*)

Bases: object

A class to manage Nexus 3 repositories.

create(*repository*)

Creates a Nexus repository with the given format and type.

Parameters **repository** (Repository) –

Returns None

delete(*name*)

Delete a repository.

Parameters **name** – name of the repository to be deleted.

get_raw_by_name(*name*)

Return the raw dict for the repository called `name`. Remember to *refresh()* to get the latest from the server.

Parameters **name** (*str*) – name of wanted repository

Returns the repository, if found.

Return type dict

Raises IndexError – if no repository named `name` is found.

raw_list()

A raw representation of the Nexus repositories.

Returns for the format, see List Repositories.

Return type dict

refresh()
Refresh local list of repositories with latest from service. A raw representation of repositories can be fetched using *raw_list()*.

nexuscli.repository.groovy.**script_create_repo**()

nexuscli.repository.validations.**is_target_supported**(*target*, *value*, *known*, *supported*)
Validate whether the a target argument is known and supported. The `target` is only used to provide a friendlier message to the user. The given `value` is checked against `known` and `supported`.

Parameters

- **target** (*str*) – name of the target, as known to the end-user.
- **value** (*str*) – value of the target key.
- **known** (*list*) – known possible values for the target.
- **supported** (*list*) – values for the target supported by us.

Raises

- `ValueError` – if given value is not in `known`.
- `NotImplementedError` – if given value is not in `supported`.

nexuscli.repository.validations.**repository_args**(*repo_type*, ***kwargs*)
Validate that the combination of arguments for a *nexuscli.repository.model.Repository* is valid.

Raises

- `ValueError` – If the value of a given argument is invalid or unsupported, or if unrecognised keyword arguments are given.
- `TypeError` – If the type of a given argument has the wrong object type.
- `NotImplementedError` – If the combination of arguments isn't yet supported.

Parameters

- **repo_type** – as given to :py:meth:‘nexuscli.nexus_repository.Repository.create()
- **kwargs** – as given to :py:meth:‘nexuscli.nexus_repository.Repository.create()

nexuscli.repository.validations.**upcase_policy_args**(*args*)
Forces upper-case on the value of the `layout_policy`, `write_policy` and `version_policy` keys of a kwargs dict.

Parameters **args** (*dict*) – kwargs given to caller

Returns a copy of the original dict with the updated values.

Return type dict

3.4 Script

class nexuscli.script.model.**Script**
Bases: `object`

class nexuscli.script.model.**ScriptCollection**(*client=None*)
Bases: `object`

A class representing a Nexus 3 script.

create(*script_dict*)

create_if_missing(*script_dict*)

delete(*script_name*)

get(*name*)

list()

run(*script_name*, *data=''*)

3.5 Util

nexuscli.nexus_util.**calculate_hash**(*hash_name*, *file_path_or_handle*)

Calculate a hash for the given file.

Parameters

- **hash_name** (*str*) – name of the hash algorithm in hashlib
- **file_path_or_handle** (*Union[str, file object]*) – source file name (str) or file handle (from open()) for the hash algorithm.

Returns the calculated hash

Return type str

nexuscli.nexus_util.**filtered_list_gen**(*raw_response*, *term=None*, *partial_match=True*)

Iterates over items yielded by raw_response_gen, validating that:

1. the *path* dict key is a str
2. the *path* value starts with starts_with (if provided)

```
>>> r = [{
>>>     'checksum': {
>>>         'md5': 'd94b865aa7620c46ef8faef7059a311c',
>>>         'sha1': '2186934d880cf24dd9ecc578335e290026695522',
>>>         'sha256': 'b7bb3424a6a6(...)4113bc38fd7807528481a8ffe3cf',
>>>         'sha512': 'e7806f3caa3e(...)3caeb9bbc54bbde286c07f837fdc'
>>>     },
>>>     'downloadUrl': 'http://nexus/repository/repo_name/a/file.ext',
>>>     'format': 'yum',
>>>     'id': 'Y2xvdWRlcmEtbWFuYWdlcj(...)mRiNWU0YjllZWQzMg',
>>>     'path': 'a/fake.rpm',
>>>     'repository': 'cloudera-manager'}]
>>>
>>> for i in filtered_list_gen(r, starts_with='a/fake.rpm')
>>>     print(i['path'])
a/fake.rpm
>>> for i in filtered_list_gen(r, starts_with='b')
>>>     print(i['path'])
# (nothing printed)
```

Parameters

- **raw_response** (*iterable*) – an iterable that yields one element of a nexus search response at a time, such as the one returned by _paginate_get().
- **term** (*str*) – if defined, only items with an attribute *path* that starts with the given parameter are returned.

- **partial_match** (*bool*) – if True, include items whose artefact path starts with the given term.

Yields *dict* – items that matched the filter.

nexuscli.nexus_util.**validate_strings**(**args*)

Checks that all given arguments have a string type (e.g. str, basestring, unicode etc)

Parameters ***args** – values to be validated.

Returns True if all arguments are of a string type. False otherwise.

Return type bool

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

n

Index

S

U

V

W