
nexus3-cli Documentation

Thiago Figueiró

Sep 14, 2021

CONTENTS:

1	Command-line Interface	1
1.1	Environment Variables	1
1.2	Commands	1
2	API	23
2.1	nexuscli	23
2.2	Examples	55
2.3	Sonatype Nexus 3 API documentation	58
3	Groovy Scripts	59
4	Indices and tables	61
	Python Module Index	63
	Index	65

COMMAND-LINE INTERFACE

Before using the CLI, you need to run *nexus3 login* to create the client configuration.

1.1 Environment Variables

The configuration configuration can be set via environment variables, which take precedence over configuration saved on file:

- NEXUS3_PASSWORD (required)
- NEXUS3_USERNAME (required)
- NEXUS3_URL (required)
- NEXUS3_API_VERSION (optional)
- NEXUS3_X509_VERIFY (optional)

The client logging level can be configured by setting an environment variable named LOG_LEVEL. Valid values are: DEBUG, INFO, WARNING (default), ERROR, CRITICAL.

1.2 Commands

After installing the `nexus3-cli` python package, the `nexus3` executable will be available on your command line. It accepts several different commands.

1.2.1 nexus3

```
nexus3 [OPTIONS] COMMAND [ARGS] ...
```

Options

--version

Show the version and exit.

blobstore

Blob store operations.

```
nexus3 blobstore [OPTIONS] COMMAND [ARGS]...
```

create

Create a new Blobstore.

```
nexus3 blobstore create [OPTIONS] COMMAND [ARGS]...
```

file

Create NAME blob store of type File stored at PATH.

```
nexus3 blobstore create file [OPTIONS] NAME PATH
```

Options

-q, --quota-type <quota_type>

Enable soft quota of the chosen type

Options spaceRemainingQuotalspaceUsedQuota

-l, --quota-limit <quota_limit>

Quota limit in bytes

Arguments

NAME

Required argument

PATH

Required argument

s3

Create NAME blob store of type S3.

```
nexus3 blobstore create s3 [OPTIONS] NAME
```

Options

- r, --bucket-region** <bucket_region>
AWS region [required]
- n, --bucket-name** <bucket_name>
S3 bucket name [required]
- k, --key-id** <key_id>
IAM access key ID [required]
- s, --secret-key** <secret_key>
Secret for key ID [required]
- bucket-prefix** <bucket_prefix>
- bucket-expiration** <bucket_expiration>
Days until deleted blobs are removed
- e, --encryption-type** <encryption_type>
S3 encryption

Options s3ManagedEncryptionkmsManagedEncryption
- E, --encryption-key** <encryption_key>
- role** <role>
IAM role to assume
- session-token** <session_token>
AWS STS session token for temporary credentials
- q, --quota-type** <quota_type>
Enable soft quota of the chosen type

Options spaceRemainingQuotalspaceUsedQuota
- l, --quota-limit** <quota_limit>
Quota limit in bytes

Arguments

NAME
Required argument

delete

Delete blob store NAME

```
nexus3 blobstore delete [OPTIONS] NAME
```

Options

--yes

Confirm the action without prompting.

Arguments

NAME

Required argument

list

List all blob stores.

```
nexus3 blobstore list [OPTIONS]
```

Options

--json, --no-json

Print output as json

show

Show the details for NAME as JSON.

```
nexus3 blobstore show [OPTIONS] NAME
```

Arguments

NAME

Required argument

cleanup-policy

Manage clean-up policies.

```
nexus3 cleanup-policy [OPTIONS] COMMAND [ARGS]...
```


create

Create or update a cleanup policy called NAME.

```
nexus3 cleanup-policy create [OPTIONS] NAME
```

Options

--format <format>

The recipe that this cleanup policy can be applied to

Options alllaptlbowerldockerlmavenlnpmnugetlpypilrawlrubygemslyum

--downloaded <downloaded>

Cleanup criteria; last downloaded in this many days.

--updated <updated>

Cleanup criteria; last updated in this many days.

--regex <regex>

Cleanup criteria; only cleanup components matching expression

--notes <notes>

Notes about your policy

Arguments

NAME

Required argument

list

```
nexus3 cleanup-policy list [OPTIONS]
```

show

Show the details for POLICY_NAME as JSON.

```
nexus3 cleanup-policy show [OPTIONS] POLICY_NAME
```

Arguments

POLICY_NAME

Required argument

delete

Recursively delete all files under `REPOSITORY_PATH`.

`REPOSITORY_PATH` must start with a repository name.

```
nexus3 delete [OPTIONS] REPOSITORY_PATH
```

Arguments

REPOSITORY_PATH

Required argument

download

Download remote `SRC` to local `DEST`. If either argument ends with a `/`, it's assumed to be a directory.

`SRC` must start with a repository name and optionally be followed by a path to be downloaded.

```
nexus3 download [OPTIONS] SRC DST
```

Options

--flatten, --no-flatten

Flatten `DEST` directory structure

--cache, --no-cache

Do not download if a local copy is already up-to-date

Arguments

SRC

Required argument

DST

Required argument

list

List all files within `REPOSITORY_PATH`.

`REPOSITORY_PATH` must start with a repository name.

```
nexus3 list [OPTIONS] REPOSITORY_PATH
```

Arguments

REPOSITORY_PATH

Required argument

login

Login to Nexus server, saving settings to `~/.nexus-cli.` and `~/.nexus-cli.env.`

```
nexus3 login [OPTIONS]
```

Options

- U, --url** <url>
Nexus OSS URL [default: `http://localhost:8081`]
- u, --username** <username>
Nexus user [default: `admin`]
- p, --password** <password>
Password for user
- x509_verify, --no-x509_verify**
Verify server certificate [default: `True`]

repository

Manage repositories.

```
nexus3 repository [OPTIONS] COMMAND [ARGS]...
```

create

Create a repository.

```
nexus3 repository create [OPTIONS] COMMAND [ARGS]...
```

group

Create a group repository.

```
nexus3 repository create group [OPTIONS] COMMAND [ARGS]...
```

recipe

Create group repository NAME.

```
nexus3 repository create group recipe [OPTIONS] NAME
```

Options

- blob-store-name** <blob_store_name>
Blobstore name to use with new repository
- strict-content, --no-strict-content**
Toggle strict content type validation
- m, --member-names** <member_names>
Repository name(s) to add to group

Arguments

NAME
Required argument

hosted

Create a hosted repository.

```
nexus3 repository create hosted [OPTIONS] COMMAND [ARGS]...
```

apt

Create a hosted apt repository.

```
nexus3 repository create hosted apt [OPTIONS] NAME
```

Options

- blob-store-name** <blob_store_name>
Blobstore name to use with new repository
- strict-content, --no-strict-content**
Toggle strict content type validation
- cleanup-policy** <cleanup_policy>
Name of existing clean-up policy to use
- write-policy** <write_policy>
Write policy to use
- Options** allowallow_ondeny
- distribution** <distribution>
Distribution to fetch; e.g.: bionic [required]

--gpg-keypair <gpg_keypair>
Path to GPG signing key [required]

--passphrase <passphrase>
Passphrase for GPG key pair

Arguments

NAME
Required argument

docker

Create a hosted docker repository.

```
nexus3 repository create hosted docker [OPTIONS] NAME
```

Options

--blob-store-name <blob_store_name>
Blobstore name to use with new repository

--strict-content, --no-strict-content
Toggle strict content type validation

--cleanup-policy <cleanup_policy>
Name of existing clean-up policy to use

--write-policy <write_policy>
Write policy to use

Options allow|allow_onceldeny

--v1-enabled, --no-v1-enabled
Enable v1 registry

--force-basic-auth, --no-force-basic-auth
Force use of basic authentication

--http-port <http_port>
Port for HTTP service

--https-port <https_port>
Port for HTTPS service

Arguments

NAME
Required argument

maven

Create a hosted maven repository.

```
nexus3 repository create hosted maven [OPTIONS] NAME
```

Options

--blob-store-name <blob_store_name>
Blobstore name to use with new repository

--strict-content, --no-strict-content
Toggle strict content type validation

--cleanup-policy <cleanup_policy>
Name of existing clean-up policy to use

--write-policy <write_policy>
Write policy to use

Options allow|allow_once|deny

--version-policy <version_policy>
Version policy to use

Options release|snapshot|mixed

--layout-policy <layout_policy>
Layout policy to use

Options strict|permissive

Arguments

NAME
Required argument

recipe

Create a hosted repository NAME of given recipe.

```
nexus3 repository create hosted recipe [OPTIONS] NAME
```

Options

--blob-store-name <blob_store_name>
Blobstore name to use with new repository

--strict-content, --no-strict-content
Toggle strict content type validation

--cleanup-policy <cleanup_policy>
Name of existing clean-up policy to use

--write-policy <write_policy>
Write policy to use

Options allowlallow_onceldeny

Arguments

NAME

Required argument

yum

Create a hosted yum repository.

```
nexus3 repository create hosted yum [OPTIONS] NAME
```

Options

--blob-store-name <blob_store_name>

Blobstore name to use with new repository

--strict-content, --no-strict-content

Toggle strict content type validation

--cleanup-policy <cleanup_policy>

Name of existing clean-up policy to use

--write-policy <write_policy>

Write policy to use

Options allowlallow_onceldeny

--depth <depth>

Depth where repodata folder(s) exist

Arguments

NAME

Required argument

proxy

Create proxy repository NAME.

```
nexus3 repository create proxy [OPTIONS] COMMAND [ARGS]...
```

apt

Create a proxy apt repository.

```
nexus3 repository create proxy apt [OPTIONS] NAME REMOTE_URL
```

Options

- blob-store-name** <blob_store_name>
Blobstore name to use with new repository
- strict-content, --no-strict-content**
Toggle strict content type validation
- cleanup-policy** <cleanup_policy>
Name of existing clean-up policy to use
- auto-block, --no-auto-block**
Disable outbound connections on remote-url access errors
- negative-cache, --no-negative-cache**
Cache responses for content missing in the remote-url
- negative-cache-ttl** <negative_cache_ttl>
Cache time in minutes
- content-max-age** <content_max_age>
Maximum age of cached artefacts
- metadata-max-age** <metadata_max_age>
Maximum age of cached artefacts metadata
- remote-auth-type** <remote_auth_type>
Only username is supported
- Options** username
- health-check, --no-health-check**
Enable Repository Health Check
- remote-username** <remote_username>
Username for remote URL
- remote-password** <remote_password>
Password for remote URL
- distribution** <distribution>
Distribution to fetch; e.g.: bionic [required]
- flat, --no-flat**
Is this repository flat?

Arguments

NAME

Required argument

REMOTE_URL

Required argument

docker

Create a docker proxy repository.

```
nexus3 repository create proxy docker [OPTIONS] NAME REMOTE_URL
```

Options

--blob-store-name <blob_store_name>

Blobstore name to use with new repository

--strict-content, --no-strict-content

Toggle strict content type validation

--cleanup-policy <cleanup_policy>

Name of existing clean-up policy to use

--auto-block, --no-auto-block

Disable outbound connections on remote-url access errors

--negative-cache, --no-negative-cache

Cache responses for content missing in the remote-url

--negative-cache-ttl <negative_cache_ttl>

Cache time in minutes

--content-max-age <content_max_age>

Maximum age of cached artefacts

--metadata-max-age <metadata_max_age>

Maximum age of cached artefacts metadata

--remote-auth-type <remote_auth_type>

Only username is supported

Options username

--health-check, --no-health-check

Enable Repository Health Check

--remote-username <remote_username>

Username for remote URL

--remote-password <remote_password>

Password for remote URL

--v1-enabled, --no-v1-enabled

Enable v1 registry

--force-basic-auth, --no-force-basic-auth

Force use of basic authentication

--http-port <http_port>
Port for HTTP service

--https-port <https_port>
Port for HTTPS service

--index-type <index_type>
Docker index type

Options registry|hub|custom

--index-url <index_url>
Required for --index-type custom

--trust-store, --no-trust-store
Required for --index-type hub or custom

Arguments

NAME
Required argument

REMOTE_URL
Required argument

maven

Create a maven proxy repository.

```
nexus3 repository create proxy maven [OPTIONS] NAME REMOTE_URL
```

Options

--blob-store-name <blob_store_name>
Blobstore name to use with new repository

--strict-content, --no-strict-content
Toggle strict content type validation

--cleanup-policy <cleanup_policy>
Name of existing clean-up policy to use

--auto-block, --no-auto-block
Disable outbound connections on remote-url access errors

--negative-cache, --no-negative-cache
Cache responses for content missing in the remote-url

--negative-cache-ttl <negative_cache_ttl>
Cache time in minutes

--content-max-age <content_max_age>
Maximum age of cached artefacts

--metadata-max-age <metadata_max_age>
Maximum age of cached artefacts metadata

--remote-auth-type <remote_auth_type>
Only username is supported

Options username

--health-check, --no-health-check
Enable Repository Health Check

--remote-username <remote_username>
Username for remote URL

--remote-password <remote_password>
Password for remote URL

--version-policy <version_policy>
Version policy to use

Options releasesnapshotmixed

--layout-policy <layout_policy>
Layout policy to use

Options strict|permissive

Arguments

NAME
Required argument

REMOTE_URL
Required argument

recipe

Create a proxy repository.

```
nexus3 repository create proxy recipe [OPTIONS] NAME REMOTE_URL
```

Options

--blob-store-name <blob_store_name>
Blobstore name to use with new repository

--strict-content, --no-strict-content
Toggle strict content type validation

--cleanup-policy <cleanup_policy>
Name of existing clean-up policy to use

--auto-block, --no-auto-block
Disable outbound connections on remote-url access errors

--negative-cache, --no-negative-cache
Cache responses for content missing in the remote-url

--negative-cache-ttl <negative_cache_ttl>
Cache time in minutes

--content-max-age <content_max_age>
Maximum age of cached artefacts

--metadata-max-age <metadata_max_age>
Maximum age of cached artefacts metadata

--remote-auth-type <remote_auth_type>
Only username is supported

Options username

--health-check, --no-health-check
Enable Repository Health Check

--remote-username <remote_username>
Username for remote URL

--remote-password <remote_password>
Password for remote URL

Arguments

NAME
Required argument

REMOTE_URL
Required argument

yum

Create a yum proxy repository.

```
nexus3 repository create proxy yum [OPTIONS] NAME REMOTE_URL
```

Options

--blob-store-name <blob_store_name>
Blobstore name to use with new repository

--strict-content, --no-strict-content
Toggle strict content type validation

--cleanup-policy <cleanup_policy>
Name of existing clean-up policy to use

--auto-block, --no-auto-block
Disable outbound connections on remote-url access errors

--negative-cache, --no-negative-cache
Cache responses for content missing in the remote-url

--negative-cache-ttl <negative_cache_ttl>
Cache time in minutes

--content-max-age <content_max_age>
Maximum age of cached artefacts

--metadata-max-age <metadata_max_age>
Maximum age of cached artefacts metadata

--remote-auth-type <remote_auth_type>
Only username is supported

Options username

--health-check, --no-health-check
Enable Repository Health Check

--remote-username <remote_username>
Username for remote URL

--remote-password <remote_password>
Password for remote URL

Arguments

NAME
Required argument

REMOTE_URL
Required argument

delete

Delete REPOSITORY_NAME (but not its blobstore).

```
nexus3 repository delete [OPTIONS] REPOSITORY_NAME
```

Options

--yes
Confirm the action without prompting.

Arguments

REPOSITORY_NAME
Required argument

list

List all repositories.

```
nexus3 repository list [OPTIONS]
```

show

Show the configuration for `REPOSITORY_NAME` as JSON.

```
nexus3 repository show [OPTIONS] REPOSITORY_NAME
```

Arguments

REPOSITORY_NAME

Required argument

script

Manage scripts.

```
nexus3 script [OPTIONS] COMMAND [ARGS]...
```

create

Create a new script called `NAME` from `FILE`.

```
nexus3 script create [OPTIONS] NAME FILE
```

Options

--script-type <script_type>

Script type

Arguments

NAME

Required argument

FILE

Required argument

delete

Delete the script called `NAME`.

```
nexus3 script delete [OPTIONS] NAME
```

Arguments

NAME

Required argument

list

List all scripts.

```
nexus3 script list [OPTIONS]
```

run

Run the script called NAME.

```
nexus3 script run [OPTIONS] NAME
```

Options

-a, --script-arguments <script_arguments>

Arguments

NAME

Required argument

security

Security operations.

```
nexus3 security [OPTIONS] COMMAND [ARGS]...
```

realm

Security realms operations.

```
nexus3 security realm [OPTIONS] COMMAND [ARGS]...
```

activate

Activate REALM_ID

```
nexus3 security realm activate [OPTIONS] REALM_ID
```

Arguments

REALM_ID

Required argument

active

List active security realms as a JSON list.

```
nexus3 security realm active [OPTIONS]
```

available

List available security realms.

```
nexus3 security realm available [OPTIONS]
```

Options

--json, --no-json

Print output as json

task

Task operations.

```
nexus3 task [OPTIONS] COMMAND [ARGS]...
```

list

List all tasks.

```
nexus3 task list [OPTIONS]
```

Options

--json, --no-json

Print output as json

run

Run TASK_ID.

```
nexus3 task run [OPTIONS] TASK_ID
```

Arguments

TASK_ID

Required argument

show

Show the details for TASK_ID as JSON.

```
nexus3 task show [OPTIONS] TASK_ID
```

Arguments

TASK_ID

Required argument

stop

Stop running TASK_ID.

```
nexus3 task stop [OPTIONS] TASK_ID
```

Arguments

TASK_ID

Required argument

upload

Upload local SRC to remote DST. If either argument ends with a /, it's assumed to be a directory.

DEST must start with a repository name and optionally be followed by the path where SRC is to be uploaded to.

```
nexus3 upload [OPTIONS] SRC DST
```

Options

- flatten, --no-flatten**
Flatten DST directory structure
- recurse, --no-recurse**
Process all SRC subdirectories

Arguments

- SRC**
Required argument
- DST**
Required argument

2.1 nexuscli

2.1.1 nexuscli package

Subpackages

nexuscli.api package

Subpackages

nexuscli.api.blobstore package

Submodules

nexuscli.api.blobstore.collection module

```
class nexuscli.api.blobstore.collection.BlobstoreCollection (nexus_http:  
                                                         nexuscli.nexus_http.NexusHttp  
                                                         = None)  
  
    Bases: nexuscli.api.base_collection.BaseCollection  
  
    create (blobstore: nexuscli.api.blobstore.model.Blobstore) → None  
  
    delete (name: str) → None  
        See Nexus 3 API documentation.  
  
    get_by_name (name: str) → nexuscli.api.blobstore.model.Blobstore  
        Get a Nexus 3 blobstore by its name.  
  
        Parameters name – name of the wanted blobstore  
  
        Returns the requested object  
  
        Raises exception.NexusClientInvalidRepository – when a repository with the  
            given name isn't found.  
  
    quota_status (name: str) → dict  
        See Nexus 3 API documentation.  
  
    raw_list () → List[str]  
        List of all blobstores on the Nexus 3 service.  
  
        Returns a list of names
```

Raises `exception.NexusClientAPIError` – if list cannot be retrieved; i.e.: any HTTP code other than 200.

nexuscli.api.blobstore.model module

class `nexuscli.api.blobstore.model.Blobstore` (*nexus_http: Optional[nexuscli.nexus_http.NexusHttp]* *Optional*
= None, ***kwargs*)

Bases: `nexuscli.api.base_model.BaseModel`

Blob store object.

Configuration example for type: File

```
{
  "softQuota": {
    "type": "spaceUsedQuota",
    "limit": 100
  },
  "name": "test",
  "type": "File",
  "path": "/tmp/test"
}
```

Configuration example for type: S3

```
{
  "name": "something-s3",
  "type": "S3",
  "bucketConfiguration": {
    "bucket": {
      "region": "us-east-1",
      "name": "mynexusbucket",
      "prefix": "path-prefix",
      "expiration": 3
    },
    "encryption": {
      "encryptionType": "s3ManagedEncryption",
      "encryptionKey": ""
    },
    "bucketSecurity": {
      "accessKeyId": "AKIASLMFS2HSCEW2MZCC",
      "role": "",
      "sessionToken": ""
    }
  }
}
```

```
ENCRYPTION_TYPES = ['s3ManagedEncryption', 'kmsManagedEncryption']
```

```
QUOTA_TYPES = ['spaceRemainingQuota', 'spaceUsedQuota']
```

```
TYPES = ['S3', 'File']
```

```
property soft_quota
```

```
property type
```

update (*params: dict*) → None

Module contents

nexuscli.api.cleanup_policy package

Submodules

nexuscli.api.cleanup_policy.collection module

class `nexuscli.api.cleanup_policy.collection.CleanupPolicyCollection` (**args*,
***kwargs*)

Bases: `nexuscli.api.base_collection.BaseCollection`

A class to manage Nexus 3 Cleanup Policies.

Parameters **client** (`nexuscli.nexus_client.NexusClient`) – the client instance that will be used to perform operations against the Nexus 3 service. You must provide this at instantiation or set it before calling any methods that require connectivity to Nexus.

GROOVY_SCRIPT_NAME = 'nexus3-cli-cleanup-policy'

Groovy script used by this class

create_or_update (*cleanup_policy: nexuscli.api.cleanup_policy.model.CleanupPolicy*)

Creates the given Cleanup Policy in the Nexus repository. If a policy with the same name already exists, it will be updated.

Parameters **cleanup_policy** – the policy to create or update.

Raises `exception.NexusClientCreateCleanupPolicyError` – when the Nexus API returns an error or unexpected result.

get_by_name (*name: str*) → `nexuscli.api.cleanup_policy.model.CleanupPolicy`

Get a Nexus 3 cleanup policy by its name.

Parameters **name** – name of the wanted policy

Returns the requested object

Raises `exception.NexusClientInvalidRepository` – when a repository with the given name isn't found.

raw_list ()

Return all cleanup policies.

Returns every policy as a list of `CleanupPolicy` instances.

Return type `list[CleanupPolicy]`

script_dependencies () → `List[str]`

This method returns any scripts that the collection class depends on. These will be installed by the `NexusClient` instance, if not already present on the server.

nexuscli.api.cleanup_policy.model module

class `nexuscli.api.cleanup_policy.model.CleanupPolicy` (*nexus_http:* *Optional[nexuscli.nexus_http.NexusHttp]* = *None*, ***kwargs*)

Bases: `nexuscli.api.base_model.BaseModel`

Represents a Nexus Cleanup Policy.

Example structure and attributes common to all repositories:

```
>>> kwargs = {
>>>     'name': 'my-policy',
>>>     'format': 'bower',
>>>     'notes': 'Some comment',
>>>     'criteria': {
>>>         'lastDownloaded': 172800,
>>>         'lastBlobUpdated': 86400,
>>>         'regex': 'matchthis'
>>>     }
>>> }
```

Parameters

- **client** (`nexuscli.nexus_client.NexusClient`) – the client instance that will be used to perform operations against the Nexus 3 service. You must provide this at instantiation or set it before calling any methods that require connectivity to Nexus.
- **name** (*str*) – name of the new policy.
- **format** (*str*) – ‘all’ or the format of the repository this policy applies to.
- **lastDownloaded** (*int*) – deletion criterion: days since artefact last downloaded
- **lastBlobUpdated** (*int*) – deletion criterion: days since last update to artefact
- **regex** (*str*) – deletion criterion: only delete artefacts that match this regular expression

Module contents

nexuscli.api.repository package

Subpackages

nexuscli.api.repository.base_models package

Submodules

nexuscli.api.repository.base_models.base_repository module

class `nexuscli.api.repository.base_models.base_repository.BaseRepository` (**args*, ***kwargs*)

Bases: `nexuscli.api.base_model.BaseModel`

The base class for Nexus repositories.

Parameters

- **name** – name of the repository.
- **recipe** (*str*) – format (recipe) of the new repository. Must be one of `RECIPES`. See Nexus documentation for details.
- **blob_store_name** (*str*) – name of an existing blob store; ‘default’ should work on most installations.
- **strict_content_type_validation** (*bool*) – Whether to validate file extension against its content type.

RECIPE_NAME = None

If a recipe is not given during initialisation, use this one as the default

TYPE = None

The repository type supported by this class

property recipe_name

The nexus3-cli name for this repository’s recipe (format). This is almost always the same as *RECIPE_NAME* with `maven` being the notable exception.

nexuscli.api.repository.base_models.group_repository module

class `nexuscli.api.repository.base_models.group_repository.GroupRepository` (**args*, ***kwargs*)

Bases: `nexuscli.api.repository.base_models.repository.Repository`

A group Nexus repository.

Parameters

- **name** (*str*) – name of the repository.
- **member_names** (*list*) – ordered name of repositories in the group
- **kwargs** – see `Repository`

TYPE = 'group'

property configuration

As per `Repository.configuration` but specific to this repository recipe and type.

Return type *str*

nexuscli.api.repository.base_models.hosted_repository module

class `nexuscli.api.repository.base_models.hosted_repository.HostedRepository` (**args*, ***kwargs*)

Bases: `nexuscli.api.repository.base_models.repository.Repository`

A hosted Nexus repository.

Parameters

- **name** (*str*) – name of the repository.
- **write_policy** (*str*) – one of `WRITE_POLICIES`. See Nexus documentation for details.
- **kwargs** – see `Repository`

TYPE = 'hosted'

WRITE_POLICIES = ['ALLOW', 'ALLOW_ONCE', 'DENY']

Nexus 3 repository write policies supported by this class.

property configuration

As per `Repository.configuration` but specific to this repository recipe and type.

Return type `str`

nexuscli.api.repository.base_models.proxy_repository module

class `nexuscli.api.repository.base_models.proxy_repository.ProxyRepository` (*args, **kwargs)

Bases: `nexuscli.api.repository.base_models.repository.Repository`

A proxy Nexus repository.

Parameters

- **name** – name of the repository.
- **remote_url** – The URL of the repository being proxied, including the protocol scheme.
- **auto_block** – Auto-block outbound connections on the repository if remote peer is detected as unreachable/unresponsive.
- **content_max_age** – How long (in minutes) to cache artifacts before rechecking the remote repository. Release repositories should use -1.
- **metadata_max_age** – How long (in minutes) to cache metadata before rechecking the remote repository.
- **negative_cache_enabled** – Cache responses for content not present in the proxied repository
- **negative_cache_ttl** – How long to cache the fact that a file was not found in the repository (in minutes)
- **kwargs** – see `Repository`

TYPE = 'proxy'

property configuration

As per `Repository.configuration` but specific to this repository recipe and type.

Return type `str`

upload (*args, **kwargs)

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

Returns number of files uploaded.

upload_directory (*args, **kwargs)

Uploads all files in a directory to the specified destination directory in this repository, honouring options flatten and recurse.

Parameters

- **source** – path to local directory to be uploaded
- **destination** – destination directory
- **recurse** (*bool*) – when True, upload directory recursively.
- **flatten** – when True, the source directory tree isn't replicated on the destination.

Returns number of files uploaded

Return type `int`

nexuscli.api.repository.base_models.repository module

class `nexuscli.api.repository.base_models.repository.Repository` (*args, **kwargs)
 Bases: `nexuscli.api.repository.base_models.base_repository.BaseRepository`

Representation of the simplest Nexus repositories.

Nexus 3 repository recipes (formats) supported by this class:

- `bower`
- `npm`
- `nuget`
- `pypi`
- `raw`
- `rubygems`
- `docker`
- `apt`

Parameters

- **name** – name of the repository.
- **nexus_client** – the *NexusClient* instance that will be used to perform operations against the Nexus 3 service. You must provide this at instantiation or set it before calling any methods that require connectivity to Nexus.
- **recipe** – format (recipe) of the new repository. Must be one of `RECIPES`. See Nexus documentation for details.
- **blob_store_name** – name of an existing blob store; 'default' should work on most installations.
- **strict_content_type_validation** – Whether to validate file extension against its content type.
- **cleanup_policy** – name of an existing repository clean-up policy.

TYPE = None

property cleanup_policy

Groovy-formatted value for the cleanup/policy attribute.

property configuration

Repository configuration represented as a python dict. The dict returned by this property can be converted to JSON for use with the `nexus3-cli-repository-create` groovy script created by the `create()` method.

Example structure and attributes common to all repositories:

```
>>> common_configuration = {
>>>     'name': 'my-repository',
>>>     'online': True,
>>>     'recipeName': 'raw',
>>>     '_state': 'present',
>>>     'attributes': {
>>>         'storage': {
>>>             'blobStoreName': 'default',
>>>         },
>>>         'cleanup': {
>>>             'policyName': None,
>>>         }
>>>     }
>>> }
```

Depending on the repository type and format (recipe), other attributes will be present.

Returns repository configuration

Return type dict

delete (*repository_path*)

Delete artefacts, recursively if `repository_path` is a directory.

Parameters `repository_path` (*str*) – location on the repository service.

Returns number of deleted files. Negative number for errors.

Return type int

download (*source*, *destination*, *flatten=False*, *nocache=False*)

Download artefacts. 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 `flatten`.

Parameters

- **source** (*str*) – location of artefact or directory on the repository service.
- **destination** (*str*) – path to the local file or directory.
- **flatten** (*bool*) – if True, the remote path isn't reproduced locally.
- **nocache** (*bool*) – if True, force download of a directory or artefact, ignoring an existing local copy. If false, it will not re-download an existing copy if its checksum matches the one in Nexus (as determined by `nexuscli.nexus_util.has_same_hash()`).

Returns number of downloaded files.

Return type `int`

download_file (*download_url*, *destination*)

Download an asset from Nexus artefact repository to local file system.

Parameters

- **download_url** (*str*) – fully-qualified URL to asset being downloaded.
- **destination** (*str*) – 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: str*) → `Iterator[Optional[str]]`

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

Parameters *repository_path* – location on the repository service.

Returns artefacts under *repository_path*.

list_raw (*repository_path: str*) → `Iterator[Dict]`

As per `list()` but yields raw Nexus artefacts as dicts.

Parameters *repository_path* – location on the repository service.

upload (*source*, *destination*, *recurse=True*, *flatten=False*)

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

Returns number of files uploaded.

upload_directory (*source*, *destination*, *recurse=True*, *flatten=False*)

Uploads all files in a directory to the specified destination directory in this repository, honouring options flatten and recurse.

Parameters

- **source** – path to local directory to be uploaded
- **destination** – destination directory
- **recurse** (*bool*) – when True, upload directory recursively.
- **flatten** – when True, the source directory tree isn't replicated on the destination.

Returns number of files uploaded

Return type `int`

upload_file (*source*, *destination*)

nexuscli.api.repository.base_models.util module

`nexuscli.api.repository.base_models.util.get_files` (*src_dir*: *Union[pathlib.Path, str]*, *recurse*: *bool = True*) → *List[pathlib.Path]*

Walks the given directory and collects files to be uploaded. If recurse option is False, only the files on the root of the directory will be returned.

Parameters

- **src_dir** – location of files
- **recurse** – If false, only the files on the root of src_dir are returned

Returns file set to be used with upload_directory

Return type `set`

Module contents

nexuscli.api.repository.recipes package

Submodules

nexuscli.api.repository.recipes.apr module

class `nexuscli.api.repository.recipes.apr.AprHostedRepository` (**args*, ***kwargs*)
Bases: `nexuscli.api.repository.recipes.apr._AprRepository`, `nexuscli.api.repository.base_models.hosted_repository.HostedRepository`

property configuration

As per `Repository.configuration` but specific to this repository recipe and type.

Return type `str`

class `nexuscli.api.repository.recipes.apr.AprProxyRepository` (**args*, ***kwargs*)
Bases: `nexuscli.api.repository.recipes.apr._AprRepository`, `nexuscli.api.repository.base_models.proxy_repository.ProxyRepository`

property configuration

As per `Repository.configuration` but specific to this repository recipe and type.

Return type `str`

class `nexuscli.api.repository.recipes.apr.AprGroupRepository`
Bases: `object`

nexuscli.api.repository.recipes.bower module

```

class nexuscli.api.repository.recipes.bower.BowerHostedRepository (*args,
                                                                    **kwargs)
    Bases: nexuscli.api.repository.recipes.bower._BowerRepository, nexuscli.api.repository.base_models.hosted_repository.HostedRepository

class nexuscli.api.repository.recipes.bower.BowerProxyRepository (*args,
                                                                    **kwargs)
    Bases: nexuscli.api.repository.recipes.bower._BowerRepository, nexuscli.api.repository.base_models.proxy_repository.ProxyRepository

class nexuscli.api.repository.recipes.bower.BowerGroupRepository (*args,
                                                                    **kwargs)
    Bases: nexuscli.api.repository.recipes.bower._BowerRepository, nexuscli.api.repository.base_models.group_repository.GroupRepository

```

nexuscli.api.repository.recipes.docker module

```

class nexuscli.api.repository.recipes.docker.DockerGroupRepository (*args,
                                                                    **kwargs)
    Bases: nexuscli.api.repository.base_models.group_repository.GroupRepository,
           nexuscli.api.repository.recipes.docker._DockerRepository

class nexuscli.api.repository.recipes.docker.DockerHostedRepository (*args,
                                                                    **kwargs)
    Bases: nexuscli.api.repository.base_models.hosted_repository.HostedRepository,
           nexuscli.api.repository.recipes.docker._DockerRepository

class nexuscli.api.repository.recipes.docker.DockerProxyRepository (*args,
                                                                    **kwargs)
    Bases: nexuscli.api.repository.base_models.proxy_repository.ProxyRepository,
           nexuscli.api.repository.recipes.docker._DockerRepository

INDEX_TYPES = ('REGISTRY', 'HUB', 'CUSTOM')

property configuration
    As per Repository.configuration but specific to this repository recipe and type.

    Return type str

```

nexuscli.api.repository.recipes.maven module

```

class nexuscli.api.repository.recipes.maven.MavenGroupRepository (*args,
                                                                    **kwargs)
    Bases: nexuscli.api.repository.base_models.group_repository.GroupRepository,
           nexuscli.api.repository.recipes.maven._MavenRepository

class nexuscli.api.repository.recipes.maven.MavenHostedRepository (*args,
                                                                    **kwargs)
    Bases: nexuscli.api.repository.base_models.hosted_repository.HostedRepository,
           nexuscli.api.repository.recipes.maven._MavenRepository

    A Maven hosted Nexus repository.

    See HostedRepository and MavenRepository

```

class `nexuscli.api.repository.recipes.maven.MavenProxyRepository` (*args, **kwargs)
 Bases: `nexuscli.api.repository.recipes.maven._MavenRepository`, `nexuscli.api.repository.base_models.proxy_repository.ProxyRepository`
 A **Maven** proxy Nexus repository.
 See `MavenRepository` and `ProxyRepository`

nexuscli.api.repository.recipes.npm module

class `nexuscli.api.repository.recipes.npm.NpmHostedRepository` (*args, **kwargs)
 Bases: `nexuscli.api.repository.recipes.npm._NpmRepository`, `nexuscli.api.repository.base_models.hosted_repository.HostedRepository`
upload_file (*src_file*, *dst_dir*=None, *dst_file*=None)
 Upload a single file to a npm repository.

Parameters

- **src_file** – path to the local file to be uploaded.
- **dst_dir** – NOT USED
- **dst_file** – NOT USED

Raises

- **exception.NexusClientInvalidRepositoryPath** – invalid repository path.
- **exception.NexusClientAPIError** – unknown response from Nexus API.

class `nexuscli.api.repository.recipes.npm.NpmProxyRepository` (*args, **kwargs)
 Bases: `nexuscli.api.repository.recipes.npm._NpmRepository`, `nexuscli.api.repository.base_models.proxy_repository.ProxyRepository`
class `nexuscli.api.repository.recipes.npm.NpmGroupRepository` (*args, **kwargs)
 Bases: `nexuscli.api.repository.recipes.npm._NpmRepository`, `nexuscli.api.repository.base_models.group_repository.GroupRepository`

nexuscli.api.repository.recipes.nuget module

class `nexuscli.api.repository.recipes.nuget.NugetHostedRepository` (*args, **kwargs)
 Bases: `nexuscli.api.repository.recipes.nuget._NugetRepository`, `nexuscli.api.repository.base_models.hosted_repository.HostedRepository`
class `nexuscli.api.repository.recipes.nuget.NugetProxyRepository` (*args, **kwargs)
 Bases: `nexuscli.api.repository.recipes.nuget._NugetRepository`, `nexuscli.api.repository.base_models.proxy_repository.ProxyRepository`
class `nexuscli.api.repository.recipes.nuget.NugetGroupRepository` (*args, **kwargs)
 Bases: `nexuscli.api.repository.recipes.nuget._NugetRepository`, `nexuscli.api.repository.base_models.group_repository.GroupRepository`

nexuscli.api.repository.recipes.pypi module

```
class nexuscli.api.repository.recipes.pypi.PyPiHostedRepository (*args,
                                                                **kwargs)
    Bases: nexuscli.api.repository.recipes.pypi._PyPiRepository, nexuscli.api.repository.base_models.hosted_repository.HostedRepository

    upload_file (src_file, dst_dir=None, dst_file=None)
        Upload a single file to a PyPI repository.

    Parameters
        • src_file – path to the local file to be uploaded.
        • dst_dir – NOT USED
        • dst_file – NOT USED

    Raises
        • exception.NexusClientInvalidRepositoryPath – invalid repository path.
        • exception.NexusClientAPIError – unknown response from Nexus API.

class nexuscli.api.repository.recipes.pypi.PyPiProxyRepository (*args,
                                                                **kwargs)
    Bases: nexuscli.api.repository.recipes.pypi._PyPiRepository, nexuscli.api.repository.base_models.proxy_repository.ProxyRepository

class nexuscli.api.repository.recipes.pypi.PyPiGroupRepository (*args,
                                                                **kwargs)
    Bases: nexuscli.api.repository.recipes.pypi._PyPiRepository, nexuscli.api.repository.base_models.group_repository.GroupRepository
```

nexuscli.api.repository.recipes.raw module

```
class nexuscli.api.repository.recipes.raw.RawHostedRepository (*args, **kwargs)
    Bases: nexuscli.api.repository.recipes.raw._RawRepository, nexuscli.api.repository.base_models.hosted_repository.HostedRepository

    upload_file (source, destination)
        Upload a single file to a raw repository.

    Parameters
        • source – path to the local file to be uploaded.
        • destination – directory under dst_repo to place file in. When None, the file is placed under the root of the raw repository

    Raises
        • exception.NexusClientInvalidRepositoryPath – invalid repository path.
        • exception.NexusClientAPIError – unknown response from Nexus API.

class nexuscli.api.repository.recipes.raw.RawProxyRepository (*args, **kwargs)
    Bases: nexuscli.api.repository.recipes.raw._RawRepository, nexuscli.api.repository.base_models.proxy_repository.ProxyRepository

class nexuscli.api.repository.recipes.raw.RawGroupRepository (*args, **kwargs)
    Bases: nexuscli.api.repository.recipes.raw._RawRepository, nexuscli.api.repository.base_models.group_repository.GroupRepository
```

nexuscli.api.repository.recipes.rubygems module

```
class nexuscli.api.repository.recipes.rubygems.RubygemsHostedRepository (*args,  
                                                                    **kwargs)  
    Bases:          nexuscli.api.repository.recipes.rubygems._RubygemsRepository,  
                    nexuscli.api.repository.base_models.hosted_repository.HostedRepository  
  
class nexuscli.api.repository.recipes.rubygems.RubygemsProxyRepository (*args,  
                                                                    **kwargs)  
    Bases:          nexuscli.api.repository.recipes.rubygems._RubygemsRepository,  
                    nexuscli.api.repository.base_models.proxy_repository.ProxyRepository  
  
class nexuscli.api.repository.recipes.rubygems.RubygemsGroupRepository (*args,  
                                                                    **kwargs)  
    Bases:          nexuscli.api.repository.recipes.rubygems._RubygemsRepository,  
                    nexuscli.api.repository.base_models.group_repository.GroupRepository
```

nexuscli.api.repository.recipes.validations module

nexuscli.api.repository.recipes.yum module

```
class nexuscli.api.repository.recipes.yum.YumGroupRepository (*args, **kwargs)  
    Bases: nexuscli.api.repository.base_models.group_repository.GroupRepository,  
            nexuscli.api.repository.recipes.yum._YumRepository  
  
class nexuscli.api.repository.recipes.yum.YumHostedRepository (*args, **kwargs)  
    Bases:          nexuscli.api.repository.base_models.hosted_repository.  
                    HostedRepository, nexuscli.api.repository.recipes.yum._YumRepository  
  
    A Yum hosted Nexus repository.  
  
    See HostedRepository and YumRepository  
  
    upload_file (source, destination)  
  
class nexuscli.api.repository.recipes.yum.YumProxyRepository (*args, **kwargs)  
    Bases: nexuscli.api.repository.base_models.proxy_repository.ProxyRepository,  
            nexuscli.api.repository.recipes.yum._YumRepository  
  
    A Yum proxy Nexus repository.  
  
    See ProxyRepository and YumRepository
```

Module contents

Submodules

nexuscli.api.repository.collection module

```
class nexuscli.api.repository.collection.RepositoryCollection (*args, **kwargs)  
    Bases: nexuscli.api.base_collection.BaseCollection  
  
    A class to manage Nexus 3 repositories.  
  
    create (repository)  
        Creates a Nexus repository with the given format and type.
```


Parameters `repository` (`Repository`) – the instance containing the settings for the repository to be created.

Raises `NexusClientCreateRepositoryError` – error creating repository.

delete (`name`)

Delete a repository.

Parameters `name` (`str`) – name of the repository to be deleted.

get_by_name (`name: str`) → `nexuscli.api.repository.base_models.repository.Repository`

Get a Nexus 3 repository by its name.

Parameters `name` (`str`) – name of the repository wanted

Return type `nexuscli.api.repository.model.Repository`

Raises `exception.NexusClientInvalidRepository` – when a repository with the given name isn't found.

get_raw_by_name (`name: str`) → `dict`

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

Parameters `name` (`str`) – name of the repository wanted

Return type `dict`

Raises `exception.NexusClientInvalidRepository` – when a repository with the given name isn't found.

new (`repo_type, **kwargs`)

Creates a `Repository` instance object

raw_list ()

A raw representation of the Nexus repositories.

Returns for the format, see [List Repositories](#).

Return type `dict`

script_dependencies () → `List[str]`

This method returns any scripts that the collection class depends on. These will be installed by the `NexusClient` instance, if not already present on the server.

set_health_check (`name: str, enable: bool = False`) → `None`

Set the health check status on a Nexus 3 repository.

Parameters

- **name** – name of the repository wanted
- **enable** – whether to enable or disable the health check

Return type `nexuscli.api.repository.model.Repository`

Raises `exception.NexusClientInvalidRepository` – when a repository with the given name isn't found.

`nexuscli.api.repository.collection.get_classes_by_type` (`repo_types: Union[str, list]`) → `List[Type[nexuscli.api.repository.base_models.repository.Repository]]`

```
nexuscli.api.repository.collection.get_repository_class (raw_configuration: dict) → Type[nexuscli.api.repository.base_models.repository]
```

Given a raw repository configuration, returns its corresponding class.

Parameters `raw_configuration` (*dict*) – configuration as returned by the SCRIPT_NAME_GET groovy script.

Returns repository class

```
nexuscli.api.repository.collection.get_repository_classes () → List[Type[nexuscli.api.repository.base_models.repository]]
```

```
nexuscli.api.repository.collection.get_supported_recipes () → List[str]
```

nexuscli.api.repository.model module

Module contents

nexuscli.api.script package

Submodules

nexuscli.api.script.model module

```
class nexuscli.api.script.model.Script (nexus_http: Optional[nexuscli.nexus_http.NexusHttp] = None, **kwargs)
```

Bases: `nexuscli.api.base_model.BaseModel`

A Class representing a Nexus 3 script.

```
class nexuscli.api.script.model.ScriptCollection (nexus_http: nexuscli.nexus_http.NexusHttp = None)
```

Bases: `nexuscli.api.base_collection.BaseCollection`

A class to manage Nexus 3 scripts.

create (*script_name*, *script_content*, *script_type*='groovy')

Create the given script in the Nexus 3 service.

Parameters

- **script_name** (*str*) – name of script to be created.
- **script_content** (*str*) – script code.
- **script_type** (*str*) – type of script to be created.

Raises `exception.NexusClientAPIError` – if the script creation isn't successful; i.e.: any HTTP code other than 204.

create_if_missing (*name*, *content*=None, *script_type*='groovy')

Creates a script in the Nexus 3 service IFF a script with the same name doesn't exist. Equivalent to checking if the script exists with `get()` and, if not, creating it with `create()`.

Parameters

- **name** (*str*) – name of script to be created.

- **content** (*Union[str, NoneType]*) – script code. If not given, the method will use `nexuscli.nexus_util.groovy_script()` to read the script code from a local file.
- **script_type** (*str*) – type of script to be created.

Raises `exception.NexusClientAPIError` – if the script creation isn't successful; i.e.: any HTTP code other than 204.

delete (*script_name*)

Deletes a script from the Nexus 3 repository.

Parameters **script_name** – name of script to be deleted.

Raises `exception.NexusClientAPIError` – if the Nexus service fails to delete the script; i.e.: any HTTP code other than 204.

exists (*name*)

Check if a script exists.

Parameters **name** – of script to verify existence.

Returns True if it exists, false otherwise

Return type `bool`

Raises `exception.NexusClientAPIError` – if the response from the Nexus service isn't recognised; i.e.: any HTTP code other than 200, 404.

get (*name*)

Get a Nexus 3 script by name.

Parameters **name** – of script to be retrieved.

Returns the script or None, if not found

Return type `dict, None`

Raises `exception.NexusClientAPIError` – if the response from the Nexus service isn't recognised; i.e.: any HTTP code other than 200, 404.

raw_list ()

run (**args, **kwargs*)

See `run_script()`

Module contents

nexuscli.api.security package

Subpackages

nexuscli.api.security.realms package

Submodules

nexuscli.api.security.realms.collection module

class `nexuscli.api.security.realms.collection.RealmCollection(*args, **kwargs)`
 Bases: `nexuscli.api.base_collection.BaseCollection`

activate (*realm_id: str*) → None

Activate a security realm.

Parameters **realm_id** – realm id to activate

Raises *exception.NexusClientAPIError* – if list cannot be retrieved; i.e.: any HTTP code

other than 204.

property active

Cached version of *raw_active()*. Use *reset()* to refresh.

property collection

Representation of *raw_list()* as objects. Use *reset()* to refresh.

raw_active () → List[str]

List of active security realms on the Nexus 3 service.

raw_list () → List[dict]

The raw Nexus server response for available security realms.

reset () → None

Clears the cached collection and causes the next call to *list()* to reload the response from the Nexus server.

nexuscli.api.security.realms.model module

class `nexuscli.api.security.realms.model.Realm` (*nexus_http: Optional[nexuscli.nexus_http.NexusHttp]* = None, **kwargs)

Bases: *nexuscli.api.base_model.BaseModel*

A Nexus 3 server security realm object.

Module contents

Module contents

nexuscli.api.task package

Submodules

nexuscli.api.task.collection module

class `nexuscli.api.task.collection.TaskCollection` (*nexus_http: nexuscli.nexus_http.NexusHttp* = None)

Bases: *nexuscli.api.base_collection.BaseCollection*

raw_list () → list

List of all script names on the Nexus 3 service.

Returns a list of names

Raises *exception.NexusClientAPIError* – if list cannot be retrieved; i.e.: any HTTP code

other than 200.

run (*task_id: str*) → None

Run a task by id

Raises *exception.NexusClientAPIError* – if task cannot be run

show (*task_id: str*) → dict

Get a single task by id

Raises *exception.NexusClientAPIError* – if task cannot be retrieved

stop (*task_id: str*) → None

Stop a running task by id

Raises *exception.NexusClientAPIError* – if task cannot be stopped

nexuscli.api.task.model module

```
class nexuscli.api.task.model.Task(nexus_http: Optional[nexuscli.nexus_http.NexusHttp] =
                                   None, **kwargs)
    Bases: nexuscli.api.base_model.BaseModel
```

Module contents

Submodules

nexuscli.api.base_collection module

```
class nexuscli.api.base_collection.BaseCollection(nexus_http:
                                                  nexuscli.nexus_http.NexusHttp
                                                  = None)
```

Bases: *object*

A base collection class that contains a Nexus 3 client.

Parameters

- **nexus_http** – the NexusHttp instance that will be used to perform requests against the Nexus
- **You must provide this at instantiation if you want to call any methods that (service.)–**
- **connectivity to Nexus. (require)–**

property list

Cached version of *raw_list()*. Use *reset()* to refresh.

raw_list () → List[dict]

reset () → None

Clears the cached collection and causes the next call to *list()* to reload the response from the Nexus server.

run_script (*script_name: str, data: Union[str, dict] = ""*)

Runs an existing script on the Nexus 3 service.

Parameters

- **script_name** – name of script to be run.
- **data** – parameters to be passed to the script, via HTTP POST. If the script being run requires a certain format or encoding, you need to prepare it yourself. Typically this is `json.dumps(data)`.

Returns the content returned by the script, if any.

Return type `str, dict`

Raises `exception.NexusClientAPIError` – if the Nexus service fails to run the script; i.e.: any HTTP code other than 200.

script_dependencies () → List[str]

This method returns any scripts that the collection class depends on. These will be installed by the NexusClient instance, if not already present on the server.

nexuscli.api.base_model module

```
class nexuscli.api.base_model.BaseModel (nexus_http: Optional[nexuscli.nexus_http.NexusHttp] = None,
                                          **kwargs)
```

Bases: `object`

Base class for Nexus 3 server objects.

Parameters

- **nexus_http** – the NexusHttp client instance that will be used to perform operations against
- **Nexus 3 service. You must provide this at instantiation if you plan on calling any (the) –**
- **that require connectivity to Nexus. (methods) –**

property configuration

The model representation as a transformed python dict suitable for converting to JSON and passing-through to the Nexus 3 API.

nexuscli.api.util module

```
nexuscli.api.util.script_for_version (script_name: str, server_version: semver.VersionInfo,
                                      versions: List[semver.VersionInfo]) → str
```

Determine if a certain nexus server version requires a different version of the given groovy script.

Parameters

- **script_name** – original name of the script.
- **server_version** – VersionInfo for the Nexus server.
- **versions** – list of VersionInfo. Each element represents an existing groovy script that must be used with server_version or greater.

Returns the version-specific name of script_name.

`nexuscli.api.util.validate_response` (*response: requests.models.Response, accepted_codes: Union[int, List[int]], exc_class: Type[Exception]* = `<class 'nexuscli.exception.NexusClientAPIError'>`) → `None`
 Ensure that `response.status_code` is one of `accepted_codes`, otherwise raise exception `exc_class` using the response content as the error message.

`nexuscli.api.util.with_min_version` (*min_version: str*) → `Callable[[F], F]`
 Verifies that the `nexus_client` instance has version greater or equal to `min_version`

nexuscli.api.validations module

`nexuscli.api.validations.ensure_known` (*target: str, value: str, known: Iterable*) → `None`
 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** – name of the target, as known to the end-user.
- **value** – value of the target key.
- **known** – known possible values for the target.

Raises **ValueError** – if given value is not in known.

Module contents

nexuscli.cli package

Submodules

nexuscli.cli.blobstore_options module

nexuscli.cli.constants module

nexuscli.cli.repository_options module

nexuscli.cli.root_commands module

Handles base/root commands (as opposed to subcommands)

`nexuscli.cli.root_commands.cmd_delete` (*nexus_client, repository_path*)
 Performs `nexus3 delete`

`nexuscli.cli.root_commands.cmd_download` (*nexus_client, src=None, dst=None, flatten=None, cache=None*)
 Performs `nexus3 download`

`nexuscli.cli.root_commands.cmd_list` (*nexus_client, full_path*)
 Performs `nexus3 list`

`nexuscli.cli.root_commands.cmd_login` (***kwargs*)
 Performs `nexus3 login`

`nexuscli.cli.root_commands.cmd_upload` (*nexus_client*, *src=None*, *dst=None*, *flatten=None*, *recurse=None*)
Performs `nexus3 upload`

nexuscli.cli.subcommand_blobstore module

`nexuscli.cli.subcommand_blobstore.cmd_create` (*nexus_client*:
nexuscli.nexus_client.NexusClient,
blob_type, *name*, ***kwargs*)
Performs `nexus3 blobstore create`

`nexuscli.cli.subcommand_blobstore.cmd_delete` (*nexus_client*:
nexuscli.nexus_client.NexusClient, *name*)

`nexuscli.cli.subcommand_blobstore.cmd_list` (*nexus_client*: *nexuscli.nexus_client.NexusClient*,
***kwargs*)
Performs `nexus3 blobstore list`

`nexuscli.cli.subcommand_blobstore.cmd_show` (*nexus_client*: *nexuscli.nexus_client.NexusClient*,
name)
Performs `nexus3 blobstore show`

nexuscli.cli.subcommand_cleanup_policy module

`nexuscli.cli.subcommand_cleanup_policy.cmd_create` (*nexus_client*:
nexuscli.nexus_client.NexusClient,
***kwargs*) → None
Performs `nexus3 cleanup_policy create`

`nexuscli.cli.subcommand_cleanup_policy.cmd_list` (*nexus_client*:
nexuscli.nexus_client.NexusClient) → None
Performs `nexus3 cleanup_policy list`

`nexuscli.cli.subcommand_cleanup_policy.cmd_show` (*nexus_client*:
nexuscli.nexus_client.NexusClient,
policy_name: str) → None
Performs `nexus3 cleanup-policy show`

nexuscli.cli.subcommand_realm module

`nexuscli.cli.subcommand_realm.cmd_activate` (*nexus_client*: *nexuscli.nexus_client.NexusClient*,
realm_id: str) → None
Performs `nexus3 security realm activate`

`nexuscli.cli.subcommand_realm.cmd_active` (*nexus_client*: *nexuscli.nexus_client.NexusClient*)
→ None
Performs `nexus3 security realm active`

`nexuscli.cli.subcommand_realm.cmd_available` (*nexus_client*:
nexuscli.nexus_client.NexusClient,
***kwargs*) → None
Performs `nexus3 security realm available`

nexuscli.cli.subcommand_repository module

`nexuscli.cli.subcommand_repository.cmd_create` (*ctx*, *repo_type=None*, ***kwargs*)
 Performs nexus3 repository create commands

`nexuscli.cli.subcommand_repository.cmd_delete` (*nexus_client*, *repository_name*)
 Performs nexus3 repository delete

`nexuscli.cli.subcommand_repository.cmd_list` (*nexus_client*)
 Performs nexus3 repository list

`nexuscli.cli.subcommand_repository.cmd_show` (*nexus_client*, *repository_name*)
 Performs `^`nexus3 repository show

nexuscli.cli.subcommand_script module

`nexuscli.cli.subcommand_script.cmd_create` (*nexus_client*, *name*, *content*, ***kwargs*)
 Performs nexus3 script create

`nexuscli.cli.subcommand_script.cmd_delete` (*nexus_client*, *name*)
 Performs nexus3 script delete

`nexuscli.cli.subcommand_script.cmd_list` (*nexus_client*)
 Performs nexus3 script list

`nexuscli.cli.subcommand_script.cmd_run` (*nexus_client*, *name*, *arguments*)
 Performs nexus3 script run

nexuscli.cli.subcommand_task module

`nexuscli.cli.subcommand_task.cmd_list` (*nexus_client*, ***kwargs*)
 Performs *nexus3 task list*

`nexuscli.cli.subcommand_task.cmd_run` (*nexus_client*, *task_id*)
 Performs *nexus3 task run*

`nexuscli.cli.subcommand_task.cmd_show` (*nexus_client*, *task_id*)
 Performs *nexus3 task show*

`nexuscli.cli.subcommand_task.cmd_stop` (*nexus_client*, *task_id*)
 Performs *nexus3 task stop*

nexuscli.cli.util module

class `nexuscli.cli.util.AliasedGroup` (**args*, ***kwargs*)
 Bases: `click_aliases.ClickAliasedGroup`

Implements execution of the first partial match for a command. Fails with a message if there are no unique matches.

See: <https://click.palletsprojects.com/en/7.x/advanced/#command-aliases>

get_command (*ctx*, *cmd_name*)
 Given a context and a command name, this returns a `Command` object if it exists or returns *None*.

add_options (*options*)

`nexuscli.cli.util.get_client()` → `nexuscli.nexus_client.NexusClient`
 Returns a Nexus Client instance. Prints a warning if the configuration file doesn't exist.

`nexuscli.cli.util.mapped_commands(command_map: dict)`

TODO: document command_map format

Parameters `command_map` –

Returns

`nexuscli.cli.util.move_to_key(mydict, new_key, keys_to_move)`

`nexuscli.cli.util.print_as_table(contents: List[Dict], fields: List)` → None

Print json API output as a table

Parameters

- **contents** – table contents
- **fields** – list of key names in contents elements to be added as columns to table

`nexuscli.cli.util.rename_keys(mydict: dict, rename_map: dict)`

`nexuscli.cli.util.upcase_values(mydict: dict, keys=[])`

`nexuscli.cli.util.with_nexus_client(click_command)`

Module contents

Submodules

nexuscli.exception module

`nexuscli.exception.API_ERROR_MAP` = {'EULA is not accepted': {'class': 'ConfigError', 'r
 Map from Nexus API response to an exception class and user-friendly error

class `nexuscli.exception.CliReturnCode`

Bases: `enum.Enum`

Error codes returned by `nexuscli.cli`

API_ERROR = 2

CAPABILITY_NOT_AVAILABLE = 6

CONFIG_ERROR = 40

CONNECTION_ERROR = 3

DOWNLOAD_ERROR = 4

INVALID_CREDENTIALS = 5

INVALID_SUBCOMMAND = 10

NOT_FOUND = 12

NOT_IMPLEMENTED = 7

NO_FILES = 1

POLICY_NOT_FOUND = 20

REPOSITORY_NOT_FOUND = 30

SUBCOMMAND_ERROR = 11

```

SUCCESS = 0
TASK_DISABLED = 13
UNKNOWN_ERROR = 99

exception nexuscli.exception.ConfigError (message: str)
    Bases: nexuscli.exception.NexusClientBaseError
    Configuration error.
    exit_code = 40

exception nexuscli.exception.DownloadError (message: str)
    Bases: nexuscli.exception.NexusClientBaseError
    Error retrieving artefact from Nexus service.
    exit_code = 4

exception nexuscli.exception.FeatureNotImplemented (message=None)
    Bases: nexuscli.exception.NexusClientBaseError
    The requested feature hasn't been implemented in this client
    exit_code = 7

exception nexuscli.exception.NexusClientAPIError (message_bytes=None)
    Bases: nexuscli.exception.NexusClientBaseError
    Unexpected response from Nexus service.
    exit_code = 2

exception nexuscli.exception.NexusClientBaseError (message: str)
    Bases: click.exceptions.ClickException
    exit_code = 99

exception nexuscli.exception.NexusClientCapabilityUnsupported (message: str)
    Bases: nexuscli.exception.NexusClientBaseError
    Client tried to use a capability that doesn't exist in this Nexus 3 server version
    exit_code = 6

exception nexuscli.exception.NexusClientConnectionError (message: str)
    Bases: nexuscli.exception.NexusClientBaseError
    Generic network connector error.
    exit_code = 3

exception nexuscli.exception.NexusClientCreateCleanupPolicyError (message:
                                                                    str)
    Bases: nexuscli.exception.NexusClientBaseError
    Used when a cleanup policy creation operation in Nexus fails.
    exit_code = 11

exception nexuscli.exception.NexusClientCreateRepositoryError (message: str)
    Bases: nexuscli.exception.NexusClientBaseError
    Used when a repository creation operation in Nexus fails.
    exit_code = 11

```

exception `nexuscli.exception.NexusClientInvalidCleanupPolicy` (*message: str*)
Bases: `nexuscli.exception.NexusClientBaseError`

The given cleanup policy does not exist in Nexus.

exit_code = 11

exception `nexuscli.exception.NexusClientInvalidCredentials` (*message: str*)
Bases: `nexuscli.exception.NexusClientBaseError`

Login credentials not accepted by Nexus service. Usually the result of a HTTP 401 response.

exit_code = 5

exception `nexuscli.exception.NexusClientInvalidRepository` (*message: str*)
Bases: `nexuscli.exception.NexusClientBaseError`

The given repository does not exist in Nexus.

exit_code = 30

exception `nexuscli.exception.NexusClientInvalidRepositoryPath` (*message: str*)
Bases: `nexuscli.exception.NexusClientBaseError`

Used when an operation against the Nexus service uses an invalid or non-existent path.

exception `nexuscli.exception.NotFound` (*message*)
Bases: `nexuscli.exception.NexusClientBaseError`

The requested object/item was not found on the server

exit_code = 12

exception `nexuscli.exception.TaskDisabled` (*message*)
Bases: `nexuscli.exception.NexusClientBaseError`

The requested task is disabled

exit_code = 13

nexuscli.nexus_client module

class `nexuscli.nexus_client.NexusClient` (*config: nexuscli.nexus_config.NexusConfig* = *None*)

Bases: `object`

A class to interact with Nexus 3's API.

Unless all keyword arguments `url`, `user` and `password` are supplied, the class will attempt to read the configuration file and, if unsuccessful, use defaults.

Parameters **config** – instance containing the configuration for the Nexus service used by this instance.

property **blobstores**

Instance of `BlobstoreCollection`

property **cleanup_policies**

Instance of `CleanupPolicyCollection`. This will automatically use the existing instance of `NexusClient` to communicate with the Nexus service.

create_scripts (*scripts: List[str]*) → *None*

property repositories

Instance of *RepositoryCollection*. This will automatically use the existing instance of *NexusClient* to communicate with the Nexus service.

property scripts

Instance of *ScriptCollection*. This will automatically use the existing instance of *NexusClient* to communicate with the Nexus service.

property security_realms

This instance uses the existing instance of *NexusClient* to communicate with the Nexus service.

property tasks

Instance of *RepositoryCollection*. This will automatically use the existing instance of *NexusClient* to communicate with the Nexus service.

nexuscli.nexus_config module

```
class nexuscli.nexus_config.NexusConfig(username='admin', password="",
                                         url='http://localhost:8081', x509_verify=True,
                                         api_version='v1', config_path=None)
```

Bases: *object*

A class to hold Nexus 3's configuration.

Unless keyword arguments `url`, `user` and `password` are supplied, the class will attempt to read the configuration file and, if unsuccessful, use defaults.

Parameters

- **username** (*str*) – username for Nexus service at given url.
- **password** (*str*) – password for username above.
- **url** (*str*) – URL to Nexus 3 OSS service.
- **x509_verify** (*bool*) – toggle certificate validation.
- **api_version** (*str*) – Nexus REST API version to be used.
- **config_path** (*str*) – local file containing configuration above in JSON format with these keys: `nexus_url`, `nexus_user`, `nexus_pass` and `nexus_verify`.

property api_version

Current API version in use.

Return type *str*

property auth

Current username and password as a tuple.

Return type *tuple*[*str*, *str*]

property config_file

Path to configuration file, as given by `config_path` during instantiation.

Return type *str*

property config_path

Path to configuration file, as given by `config_path` during instantiation.

dump () → *str*

Writes the current configuration to disk under property: `config_file`.

If a file already exists, it will be overwritten. The permission will be set to read/write to the owner only.

return: name of the file written

dump_env()

Writes the current configuration to disk under property:*config_file*.env.

If a file already exists, it will be overwritten. The permission will be set to read/write to the owner only.

return: name of the file written

load()

Load the configuration settings from the file specified by *config_file*.

The configuration file is in JSON format and expects these keys: *nexus_user*, *nexus_pass*, *nexus_url*, *nexus_verify*.

property to_dict

Current instance configuration.

Return type *dict*

property url

The Nexus service URL

Return type *str*

property x509_verify

Whether to validate the x509 certificate when using https to access the Nexus service

Return type *str*

nexuscli.nexus_http module

class `nexuscli.nexus_http.NexusHttp` (*config: nexuscli.nexus_config.NexusConfig = None*)

Bases: *object*

get (*endpoint: str, **kwargs*) → *requests.models.Response*

Performs a HTTP GET request on the given endpoint.

Parameters *endpoint* – name of the Nexus REST API endpoint.

request (*method: str, endpoint: str, service_url: Optional[str] = None, **kwargs*) → *requests.models.Response*

Performs a HTTP request to the Nexus REST API on the specified endpoint.

Parameters

- **method** – one of *get*, *put*, *post*, *delete*.
- **endpoint** – URI path to be appended to the service URL.
- **service_url** – override the default URL to use for the request, which is created by joining *rest_url* and *endpoint*.
- **kwargs** – as per *requests.request()*.

property rest_url

Full URL to the Nexus REST API, based on the *url* and *version* from *config*.

Return type *str*

property server_version

Parse the Server header from a Nexus request response and return as version information. The method expects the header *Server* to be present and formatted as, e.g., 'Nexus/3.19.1-01 (OSS)'

Returns the parsed version. If it can't be determined, return *None*.

Return type Union[None,semver.VersionInfo]

property service_url

Full URL to the Nexus REST API, based on the url and version from config.

Return type str

nexuscli.nexus_util module

nexuscli.nexus_util.REMOTE_PATH_SEPARATOR = '/'

The character used by the Nexus server as a path separator

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** (str) – source file name (str) or file handle (file-like) for the hash algorithm.

Returns the calculated hash

Return type str

nexuscli.nexus_util.ensure_exists(path: pathlib.Path, is_dir: bool = False)

Ensures a path exists.

Parameters

- **path** (pathlib.Path) – the path to ensure
- **is_dir** (bool) – whether the path is a directory.

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.get_dst_path_and_file(source: str, destination: str) → Tuple[str, str]`

Given a source file and destination path, return a destination path and destination file name.

Parameters

- **source** – the source file path. The name of this file is used as the returned file name when the given destination is a directory.
- **destination** – the destination path. When this is a path to a file, the returned destination is the base path of this value and the returned file name matches this one.

`nexuscli.nexus_util.groovy_script(script_name)`

Returns the content for a groovy script located in the package installation path under `script/groovy`.

E.g.: `groovy_script('foo')` returns the content for the file at `.../site-packages/nexuscli/script/groovy/foo.groovy`.

Parameters **script_name** – file name of the groovy script; `.groovy` is appended to this parameter to form the file name.

Returns content for the groovy script

Return type `str`

`nexuscli.nexus_util.has_same_hash(artefact, filepath)`

Checks if a Nexus artefact has the same hash as a local filepath.

Parameters

- **artefact** (*dict*) – as returned by `list_raw()`
- **filepath** – local file path

Returns True if artefact and filepath have the same hash.

Return type `bool`

`nexuscli.nexus_util.pop_repository(component_path)`

Helper for `split_component_path`. Returns the repository and the remainder of the `component_path` as a `path_fragments` list.

Parameters **component_path** – the component path, as given to `split_component_path`.

Returns tuple of (repository, path_fragments)

Return type `tuple(str, list)`

`nexuscli.nexus_util.remote_path_to_local(remote_src, local_dst, flatten, create=True)`

Takes the remote path of an asset (without the repository name), the desired destination in the local file system, and creates the fully qualified path according to the instance settings.

If `self.flatten` is True, the `remote_path` isn't reproduced locally.

If the remote is a directory, we'll always assume the destination is also a directory, even if it doesn't end with a `/`.

Parameters

- **remote_src** – path to the artefact as reported by the artefact service (i.e.: the *path* attribute of an asset object).
- **local_dst** – desired location in the local filesystem for the remote_path.
- **create** – whether or not to create the local destination file or directory.

Returns the local path to be used.

`nexuscli.nexus_util.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 `/` 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, then the filename is set to `None`.

Return type `tuple[str, str, str]`

`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`

Module contents

```
class nexuscli.NexusConfig (username='admin', password="", url='http://localhost:8081',
                             x509_verify=True, api_version='v1', config_path=None)
```

Bases: `object`

A class to hold Nexus 3's configuration.

Unless keyword arguments `url`, `user` and `password` are supplied, the class will attempt to read the configuration file and, if unsuccessful, use defaults.

Parameters

- **username** (*str*) – username for Nexus service at given url.
- **password** (*str*) – password for username above.
- **url** (*str*) – URL to Nexus 3 OSS service.
- **x509_verify** (*bool*) – toggle certificate validation.
- **api_version** (*str*) – Nexus REST API version to be used.
- **config_path** (*str*) – local file containing configuration above in JSON format with these keys: `nexus_url`, `nexus_user`, `nexus_pass` and `nexus_verify`.

property api_version

Current API version in use.

Return type *str*

property auth

Current username and password as a tuple.

Return type *tuple*[*str*, *str*]

property config_file

Path to configuration file, as given by `config_path` during instantiation.

Return type *str*

property config_path

Path to configuration file, as given by `config_path` during instantiation.

dump () → *str*

Writes the current configuration to disk under property:*config_file*.

If a file already exists, it will be overwritten. The permission will be set to read/write to the owner only.

return: name of the file written

dump_env ()

Writes the current configuration to disk under property:*config_file.env*.

If a file already exists, it will be overwritten. The permission will be set to read/write to the owner only.

return: name of the file written

load ()

Load the configuration settings from the file specified by *config_file*.

The configuration file is in JSON format and expects these keys: `nexus_user`, `nexus_pass`, `nexus_url`, `nexus_verify`.

property to_dict

Current instance configuration.

Return type `dict`

property url

The Nexus service URL

Return type `str`

property x509_verify

Whether to validate the x509 certificate when using https to access the Nexus service

Return type `str`

class `nexuscli.NexusClient` (*config: nexuscli.nexus_config.NexusConfig = None*)

Bases: `object`

A class to interact with Nexus 3's API.

Unless all keyword arguments `url`, `user` and `password` are supplied, the class will attempt to read the configuration file and, if unsuccessful, use defaults.

Parameters `config` – instance containing the configuration for the Nexus service used by this instance.

property blobstores

Instance of `BlobstoreCollection`

property cleanup_policies

Instance of `CleanupPolicyCollection`. This will automatically use the existing instance of `NexusClient` to communicate with the Nexus service.

create_scripts (*scripts: List[str]*) → `None`

property repositories

Instance of `RepositoryCollection`. This will automatically use the existing instance of `NexusClient` to communicate with the Nexus service.

property scripts

Instance of `ScriptCollection`. This will automatically use the existing instance of `NexusClient` to communicate with the Nexus service.

property security_realms

This instance uses the existing instance of `NexusClient` to communicate with the Nexus service.

property tasks

Instance of `RepositoryCollection`. This will automatically use the existing instance of `NexusClient` to communicate with the Nexus service.

2.2 Examples

Here are some basic operations to get you started. Every command available on the CLI uses this API interface, so the source code in `src/nexuscli/nexus_client.py` and `src/nexuscli/cli/` is a good source of examples.

In all examples below you will need to instantiate a client:

```
import nexuscli, pprint
nexus_config = nexuscli.nexus_config.NexusConfig()
nexus_config.load() # read config from ~/.nexus-cli
nexus_client = nexuscli.nexus_client.NexusClient(config=nexus_config)
```

2.2.1 Get list of repositories

```
nexus_client.repositories.list
[{'attributes': {},
 'format': 'maven2',
 'name': 'maven-snapshots',
 'type': 'hosted',
 'url': 'http://localhost:8081/repository/maven-snapshots'},
 {'attributes': {'proxy': {'remoteUrl': 'https://repo1.maven.org/maven2/'}},
 'format': 'maven2',
 'name': 'maven-central',
 'type': 'proxy',
 'url': 'http://localhost:8081/repository/maven-central'},
 {'attributes': {},
 'format': 'nuget',
 'name': 'nuget-group',
 'type': 'group',
 'url': 'http://localhost:8081/repository/nuget-group'},
 {'attributes': {'proxy': {'remoteUrl': 'https://www.nuget.org/api/v2/'}},
 'format': 'nuget',
 'name': 'nuget.org-proxy',
 'type': 'proxy',
 'url': 'http://localhost:8081/repository/nuget.org-proxy'},
 {'attributes': {},
 'format': 'maven2',
 'name': 'maven-releases',
 'type': 'hosted',
 'url': 'http://localhost:8081/repository/maven-releases'},
 {'attributes': {},
 'format': 'nuget',
 'name': 'nuget-hosted',
 'type': 'hosted',
 'url': 'http://localhost:8081/repository/nuget-hosted'},
 {'attributes': {},
 'format': 'maven2',
 'name': 'maven-public',
 'type': 'group',
 'url': 'http://localhost:8081/repository/maven-public'}]
```

2.2.2 Load a server repository into a local object

```
r = nexus_client.repositories.get_by_name('maven-central')
r.cleanup_policy
[None]
r.strict_content
False
r.version_policy
'RELEASE'
```

2.2.3 Create a repository

```
from nexuscli.api.repository.model import RawHostedRepository

r = RawHostedRepository(
    name='my-repository',
    blob_store_name='default',
    strict_content_type_validation=False,
    write_policy='ALLOW',
)

nexus_client.repositories.create(r)

nexus_client.repositories.get_raw_by_name('my-repository')
{'name': 'my-repository',
 'format': 'raw',
 'type': 'hosted',
 'url': 'http://localhost:8081/repository/my-repository'}
```

2.2.4 Delete a repository

```
nexus_client.repositories.delete('my-repository')

nexus_client.repositories.get_raw_by_name('my-repository')

NexusClientInvalidRepository: my-repository
```

2.2.5 Upload a file

```
repository = nexus_client.repositories.get_by_name('my-repository')
repository.upload('/etc/passwd', '/etc/passwd')
1
list(repository.list('/'))
['etc/passwd']
```

2.2.6 Upload a directory

```
repository.upload_directory('src/nexuscli/api', '/thats-a-lot-of-files')
125
len(list(repository.list('/')))
125
list(repository.list('/'))[:5]
['thats-a-lot-of-files/util.py',
 'thats-a-lot-of-files/__init__.py',
 'thats-a-lot-of-files/base_collection.py',
 'thats-a-lot-of-files/base_model.py',
 'thats-a-lot-of-files/repository/__init__.py']
```

2.3 Sonatype Nexus 3 API documentation

- [REST API](#)
- [Script API](#)

GROOVY SCRIPTS

This package makes use of Groovy scripts to perform actions that are not available through the Nexus 3 REST API. All scripts added have names starting with `nexus3-cli-`.

```
$ nexus3 script list
```

Name	Type	Content

↪-----		
nexus3-cli-cleanup-policy_3.27.0	groovy	// Original from: // https://github.com/...

↪-----		
nexus3-cli-repository-delete	groovy	log.info("Deleting <repository=\${args}> ↪"...

↪-----		
nexus3-cli-repository-get	groovy	import groovy.json.JsonBuilder Boolean ...

↪-----		
nexus3-cli-repository-create_3.21.0	groovy	import groovy.json.JsonSlurper import or...

You can delete them all by running:

```
$ nexus3 script list | grep ^nexus3- | awk '{ print $1 }' | xargs --no-run-if-empty -  
↪n1 nexus3 script del  
Name (type)
```

Note: On macOS, remove the `--no-run-if-empty` as it's not supported by the BSD `xargs`.

To increase verbosity of logging for the scripts, create a new logger (e.g.: <http://localhost:8081/#admin/support/logging>) with logger name `org.sonatype.nexus.script.plugin.internal.rest.ScriptResource` and logging level `DEBUG` or `TRACE`.

INDICES AND TABLES

- [genindex](#)
- [modindex](#)
- [search](#)

See also: [Documentation for the Sonatype Nexus Repository Manager 3](#).

PYTHON MODULE INDEX

n

nexuscli, 54
nexuscli.api, 43
nexuscli.api.base_collection, 41
nexuscli.api.base_model, 42
nexuscli.api.blobstore, 25
nexuscli.api.blobstore.collection, 23
nexuscli.api.blobstore.model, 24
nexuscli.api.cleanup_policy, 26
nexuscli.api.cleanup_policy.collection, 25
nexuscli.api.cleanup_policy.model, 26
nexuscli.api.repository, 38
nexuscli.api.repository.base_models, 32
nexuscli.api.repository.base_models.base_repository, 26
nexuscli.api.repository.base_models.group_repository, 27
nexuscli.api.repository.base_models.hosted_repository, 27
nexuscli.api.repository.base_models.proxy_repository, 28
nexuscli.api.repository.base_models.repository, 29
nexuscli.api.repository.base_models.util, 32
nexuscli.api.repository.collection, 36
nexuscli.api.repository.model, 38
nexuscli.api.repository.recipes, 36
nexuscli.api.repository.recipes.apr, 32
nexuscli.api.repository.recipes.bower, 33
nexuscli.api.repository.recipes.docker, 33
nexuscli.api.repository.recipes.maven, 33
nexuscli.api.repository.recipes.npm, 34
nexuscli.api.repository.recipes.nuget, 34
nexuscli.api.repository.recipes.pypi, 35
nexuscli.api.repository.recipes.raw, 35
nexuscli.api.repository.recipes.rubygems, 36
nexuscli.api.repository.recipes.validations, 36
nexuscli.api.repository.recipes.yum, 36
nexuscli.api.script, 39
nexuscli.api.script.model, 38
nexuscli.api.security, 40
nexuscli.api.security.realms, 40
nexuscli.api.security.realms.collection, 39
nexuscli.api.security.realms.model, 40
nexuscli.api.task, 41
nexuscli.api.task.collection, 40
nexuscli.api.task.model, 41
nexuscli.api.util, 42
nexuscli.api.validations, 43
nexuscli.cli, 46
nexuscli.cli.blobstore_options, 43
nexuscli.cli.constants, 43
nexuscli.cli.repository_options, 43
nexuscli.cli.root_commands, 43
nexuscli.cli.subcommand_blobstore, 44
nexuscli.cli.subcommand_cleanup_policy, 44
nexuscli.cli.subcommand_realm, 44
nexuscli.cli.subcommand_repository, 45
nexuscli.cli.subcommand_script, 45
nexuscli.cli.subcommand_task, 45
nexuscli.cli.util, 45
nexuscli.exception, 46
nexuscli.nexus_client, 48
nexuscli.nexus_config, 49
nexuscli.nexus_http, 50
nexuscli.nexus_util, 51

INDEX

Symbols

-E
 nexus3-blobstore-create-s3 command
 line option, 3

-U
 nexus3-login command line option, 7

--auto-block
 nexus3-repository-create-proxy-apt
 command line option, 12
 nexus3-repository-create-proxy-docker
 command line option, 13
 nexus3-repository-create-proxy-maven
 command line option, 14
 nexus3-repository-create-proxy-recipe
 command line option, 15
 nexus3-repository-create-proxy-yum
 command line option, 16

--blob-store-name <blob_store_name>
 nexus3-repository-create-group-recipe
 command line option, 8
 nexus3-repository-create-hosted-apt
 command line option, 8
 nexus3-repository-create-hosted-docker
 command line option, 9
 nexus3-repository-create-hosted-maven
 command line option, 10
 nexus3-repository-create-hosted-recipe
 command line option, 10
 nexus3-repository-create-hosted-yum
 command line option, 11
 nexus3-repository-create-proxy-apt
 command line option, 12
 nexus3-repository-create-proxy-docker
 command line option, 13
 nexus3-repository-create-proxy-maven
 command line option, 14
 nexus3-repository-create-proxy-recipe
 command line option, 15
 nexus3-repository-create-proxy-yum
 command line option, 16

--bucket-expiration
 <bucket_expiration>
 nexus3-blobstore-create-s3 command
 line option, 3

--bucket-name <bucket_name>
 nexus3-blobstore-create-s3 command
 line option, 3

--bucket-prefix <bucket_prefix>
 nexus3-blobstore-create-s3 command
 line option, 3

--bucket-region <bucket_region>
 nexus3-blobstore-create-s3 command
 line option, 3

--cache
 nexus3-download command line
 option, 6

--cleanup-policy <cleanup_policy>
 nexus3-repository-create-hosted-apt
 command line option, 8
 nexus3-repository-create-hosted-docker
 command line option, 9
 nexus3-repository-create-hosted-maven
 command line option, 10
 nexus3-repository-create-hosted-recipe
 command line option, 10
 nexus3-repository-create-hosted-yum
 command line option, 11
 nexus3-repository-create-proxy-apt
 command line option, 12
 nexus3-repository-create-proxy-docker
 command line option, 13
 nexus3-repository-create-proxy-maven
 command line option, 14
 nexus3-repository-create-proxy-recipe
 command line option, 15
 nexus3-repository-create-proxy-yum
 command line option, 16

--content-max-age <content_max_age>
 nexus3-repository-create-proxy-apt
 command line option, 12
 nexus3-repository-create-proxy-docker
 command line option, 13
 nexus3-repository-create-proxy-maven
 command line option, 14

```

nexus3-repository-create-proxy-recipe
  command line option, 15
nexus3-repository-create-proxy-yum
  command line option, 16
--depth <depth>
  nexus3-repository-create-hosted-yum
    command line option, 11
--distribution <distribution>
  nexus3-repository-create-hosted-apt
    command line option, 8
  nexus3-repository-create-proxy-apt
    command line option, 12
--downloaded <downloaded>
  nexus3-cleanup-policy-create
    command line option, 5
--encryption-key <encryption_key>
  nexus3-blobstore-create-s3 command
    line option, 3
--encryption-type <encryption_type>
  nexus3-blobstore-create-s3 command
    line option, 3
--flat
  nexus3-repository-create-proxy-apt
    command line option, 12
--flatten
  nexus3-download command line
    option, 6
  nexus3-upload command line option,
    22
--force-basic-auth
  nexus3-repository-create-hosted-docker
    command line option, 9
  nexus3-repository-create-proxy-docker
    command line option, 13
--format <format>
  nexus3-cleanup-policy-create
    command line option, 5
--gpg-keypair <gpg_keypair>
  nexus3-repository-create-hosted-apt
    command line option, 8
--health-check
  nexus3-repository-create-proxy-apt
    command line option, 12
  nexus3-repository-create-proxy-docker
    command line option, 13
  nexus3-repository-create-proxy-maven
    command line option, 15
  nexus3-repository-create-proxy-recipe
    command line option, 16
  nexus3-repository-create-proxy-yum
    command line option, 17
--http-port <http_port>
  nexus3-repository-create-hosted-docker
    command line option, 9
  nexus3-repository-create-proxy-docker
    command line option, 13
--https-port <https_port>
  nexus3-repository-create-hosted-docker
    command line option, 9
  nexus3-repository-create-proxy-docker
    command line option, 14
--index-type <index_type>
  nexus3-repository-create-proxy-docker
    command line option, 14
--index-url <index_url>
  nexus3-repository-create-proxy-docker
    command line option, 14
--json
  nexus3-blobstore-list command line
    option, 4
  nexus3-security-realm-available
    command line option, 20
  nexus3-task-list command line
    option, 20
--key-id <key_id>
  nexus3-blobstore-create-s3 command
    line option, 3
--layout-policy <layout_policy>
  nexus3-repository-create-hosted-maven
    command line option, 10
  nexus3-repository-create-proxy-maven
    command line option, 15
--member-names <member_names>
  nexus3-repository-create-group-recipe
    command line option, 8
--metadata-max-age <metadata_max_age>
  nexus3-repository-create-proxy-apt
    command line option, 12
  nexus3-repository-create-proxy-docker
    command line option, 13
  nexus3-repository-create-proxy-maven
    command line option, 14
  nexus3-repository-create-proxy-recipe
    command line option, 16
  nexus3-repository-create-proxy-yum
    command line option, 16
--negative-cache
  nexus3-repository-create-proxy-apt
    command line option, 12
  nexus3-repository-create-proxy-docker
    command line option, 13
  nexus3-repository-create-proxy-maven
    command line option, 14
  nexus3-repository-create-proxy-recipe
    command line option, 15
  nexus3-repository-create-proxy-yum
    command line option, 16
--negative-cache-ttl

```

```

    <negative_cache_ttl>
nexus3-repository-create-proxy-apt
    command line option, 12
nexus3-repository-create-proxy-docker
    command line option, 13
nexus3-repository-create-proxy-maven
    command line option, 14
nexus3-repository-create-proxy-recipe
    command line option, 15
nexus3-repository-create-proxy-yum
    command line option, 16
--no-auto-block
    nexus3-repository-create-proxy-apt
        command line option, 12
    nexus3-repository-create-proxy-docker
        command line option, 13
    nexus3-repository-create-proxy-maven
        command line option, 14
    nexus3-repository-create-proxy-recipe
        command line option, 15
    nexus3-repository-create-proxy-yum
        command line option, 16
--no-cache
    nexus3-download command line
        option, 6
--no-flat
    nexus3-repository-create-proxy-apt
        command line option, 12
--no-flatten
    nexus3-download command line
        option, 6
    nexus3-upload command line option,
        22
--no-force-basic-auth
    nexus3-repository-create-hosted-docker
        command line option, 9
    nexus3-repository-create-proxy-docker
        command line option, 13
--no-health-check
    nexus3-repository-create-proxy-apt
        command line option, 12
    nexus3-repository-create-proxy-docker
        command line option, 13
    nexus3-repository-create-proxy-maven
        command line option, 15
    nexus3-repository-create-proxy-recipe
        command line option, 16
    nexus3-repository-create-proxy-yum
        command line option, 17
--no-json
    nexus3-blobstore-list command line
        option, 4
    nexus3-security-realm-available
        command line option, 20
    nexus3-task-list command line
        option, 20
--no-negative-cache
    nexus3-repository-create-proxy-apt
        command line option, 12
    nexus3-repository-create-proxy-docker
        command line option, 13
    nexus3-repository-create-proxy-maven
        command line option, 14
    nexus3-repository-create-proxy-recipe
        command line option, 15
    nexus3-repository-create-proxy-yum
        command line option, 16
--no-recurse
    nexus3-upload command line option,
        22
--no-strict-content
    nexus3-repository-create-group-recipe
        command line option, 8
    nexus3-repository-create-hosted-apt
        command line option, 8
    nexus3-repository-create-hosted-docker
        command line option, 9
    nexus3-repository-create-hosted-maven
        command line option, 10
    nexus3-repository-create-hosted-recipe
        command line option, 10
    nexus3-repository-create-hosted-yum
        command line option, 11
    nexus3-repository-create-proxy-apt
        command line option, 12
    nexus3-repository-create-proxy-docker
        command line option, 13
    nexus3-repository-create-proxy-maven
        command line option, 14
    nexus3-repository-create-proxy-recipe
        command line option, 15
    nexus3-repository-create-proxy-yum
        command line option, 16
--no-trust-store
    nexus3-repository-create-proxy-docker
        command line option, 14
--no-v1-enabled
    nexus3-repository-create-hosted-docker
        command line option, 9
    nexus3-repository-create-proxy-docker
        command line option, 13
--no-x509_verify
    nexus3-login command line option, 7
--notes <notes>
    nexus3-cleanup-policy-create
        command line option, 5
--passphrase <passphrase>
    nexus3-repository-create-hosted-apt

```

```

        command line option,9
--password <password>
    nexus3-login command line option,7
--quota-limit <quota_limit>
    nexus3-blobstore-create-file
        command line option,2
    nexus3-blobstore-create-s3 command
        line option,3
--quota-type <quota_type>
    nexus3-blobstore-create-file
        command line option,2
    nexus3-blobstore-create-s3 command
        line option,3
--recurse
    nexus3-upload command line option,
        22
--regex <regex>
    nexus3-cleanup-policy-create
        command line option,5
--remote-auth-type <remote_auth_type>
    nexus3-repository-create-proxy-apt
        command line option,12
    nexus3-repository-create-proxy-docker
        command line option,13
    nexus3-repository-create-proxy-maven
        command line option,14
    nexus3-repository-create-proxy-recipe
        command line option,16
    nexus3-repository-create-proxy-yum
        command line option,17
--remote-password <remote_password>
    nexus3-repository-create-proxy-apt
        command line option,12
    nexus3-repository-create-proxy-docker
        command line option,13
    nexus3-repository-create-proxy-maven
        command line option,15
    nexus3-repository-create-proxy-recipe
        command line option,16
    nexus3-repository-create-proxy-yum
        command line option,17
--remote-username <remote_username>
    nexus3-repository-create-proxy-apt
        command line option,12
    nexus3-repository-create-proxy-docker
        command line option,13
    nexus3-repository-create-proxy-maven
        command line option,15
    nexus3-repository-create-proxy-recipe
        command line option,16
    nexus3-repository-create-proxy-yum
        command line option,17
--role <role>
    nexus3-blobstore-create-s3 command
        line option,3
    nexus3-repository-create-group-recipe
        command line option,8
    nexus3-repository-create-hosted-apt
        command line option,8
    nexus3-repository-create-hosted-docker
        command line option,9
    nexus3-repository-create-hosted-maven
        command line option,10
    nexus3-repository-create-hosted-recipe
        command line option,10
    nexus3-repository-create-hosted-yum
        command line option,11
    nexus3-repository-create-proxy-apt
        command line option,12
    nexus3-repository-create-proxy-docker
        command line option,13
    nexus3-repository-create-proxy-maven
        command line option,14
    nexus3-repository-create-proxy-recipe
        command line option,15
    nexus3-repository-create-proxy-yum
        command line option,16
--script-arguments <script_arguments>
    nexus3-script-run command line
        option,19
--script-type <script_type>
    nexus3-script-create command line
        option,18
--secret-key <secret_key>
    nexus3-blobstore-create-s3 command
        line option,3
--session-token <session_token>
    nexus3-blobstore-create-s3 command
        line option,3
--strict-content
    nexus3-repository-create-group-recipe
        command line option,8
    nexus3-repository-create-hosted-apt
        command line option,8
    nexus3-repository-create-hosted-docker
        command line option,9
    nexus3-repository-create-hosted-maven
        command line option,10
    nexus3-repository-create-hosted-recipe
        command line option,10
    nexus3-repository-create-hosted-yum
        command line option,11
    nexus3-repository-create-proxy-apt
        command line option,12
    nexus3-repository-create-proxy-docker
        command line option,13
    nexus3-repository-create-proxy-maven
        command line option,14
    nexus3-repository-create-proxy-recipe
        command line option,15
    nexus3-repository-create-proxy-yum
        command line option,16
--trust-store
    nexus3-repository-create-proxy-docker
        command line option,14
--updated <updated>
    nexus3-cleanup-policy-create
        command line option,5
--url <url>
    nexus3-login command line option,7
--username <username>
    nexus3-login command line option,7
--v1-enabled
    nexus3-repository-create-hosted-docker
        command line option,9
    nexus3-repository-create-proxy-docker
        command line option,13
--version
    nexus3 command line option,2
--version-policy <version_policy>

```


nexus3-repository-create-hosted-maven	line option, 3
command line option, 10	-u
nexus3-repository-create-proxy-maven	nexus3-login command line option, 7
command line option, 15	
--write-policy <write_policy>	
nexus3-repository-create-hosted-apt	activate() (nexuscli.api.security.realms.collection.RealmCollection method), 39
command line option, 8	
nexus3-repository-create-hosted-docker	active() (nexuscli.api.security.realms.collection.RealmCollection property), 40
command line option, 9	
nexus3-repository-create-hosted-maven	add_options() (in module nexuscli.cli.util), 45
command line option, 10	AliasedGroup (class in nexuscli.cli.util), 45
nexus3-repository-create-hosted-recipe	API_ERROR (nexuscli.exception.CliReturnCode attribute), 46
command line option, 10	
nexus3-repository-create-hosted-yum	API_ERROR_MAP (in module nexuscli.exception), 46
command line option, 11	
--x509_verify	api_version() (nexuscli.nexus_config.NexusConfig property), 49
nexus3-login command line option, 7	api_version() (nexuscli.NexusConfig property), 54
--yes	AptGroupRepository (class in nexuscli.api.repository.recipes.apt), 32
nexus3-blobstore-delete command	AptHostedRepository (class in nexuscli.api.repository.recipes.apt), 32
line option, 4	AptProxyRepository (class in nexuscli.api.repository.recipes.apt), 32
nexus3-repository-delete command	auth() (nexuscli.nexus_config.NexusConfig property), 49
line option, 17	auth() (nexuscli.NexusConfig property), 54
-a	
nexus3-script-run command line	
option, 19	
-e	
nexus3-blobstore-create-s3 command	
line option, 3	
-k	
nexus3-blobstore-create-s3 command	
line option, 3	
-l	
nexus3-blobstore-create-file	
command line option, 2	
nexus3-blobstore-create-s3 command	
line option, 3	
-m	
nexus3-repository-create-group-recipe	
command line option, 8	
-n	
nexus3-blobstore-create-s3 command	
line option, 3	
-p	
nexus3-login command line option, 7	
-q	
nexus3-blobstore-create-file	
command line option, 2	
nexus3-blobstore-create-s3 command	
line option, 3	
-r	
nexus3-blobstore-create-s3 command	
line option, 3	
-s	
nexus3-blobstore-create-s3 command	

A

B

C

[cleanup_policies\(\)](#) ([nexuscli.nexus_client.NexusClient](#) property), [48](#)
[cleanup_policies\(\)](#) ([nexuscli.NexusClient](#) property), [55](#)
[cleanup_policy\(\)](#) ([nexuscli.api.repository.base_models.repository.Repository](#) subcommand_cleanup_policy), [30](#)
[CleanupPolicy](#) (class in [nexuscli.api.cleanup_policy.model](#)), [26](#)
[CleanupPolicyCollection](#) (class in [nexuscli.api.cleanup_policy.collection](#)), [25](#)
[CliReturnCode](#) (class in [nexuscli.exception](#)), [46](#)
[cmd_activate\(\)](#) (in module [nexuscli.cli.subcommand_realm](#)), [44](#)
[cmd_active\(\)](#) (in module [nexuscli.cli.subcommand_realm](#)), [44](#)
[cmd_available\(\)](#) (in module [nexuscli.cli.subcommand_realm](#)), [44](#)
[cmd_create\(\)](#) (in module [nexuscli.cli.subcommand_blobstore](#)), [44](#)
[cmd_create\(\)](#) (in module [nexuscli.cli.subcommand_cleanup_policy](#)), [44](#)
[cmd_create\(\)](#) (in module [nexuscli.cli.subcommand_repository](#)), [45](#)
[cmd_create\(\)](#) (in module [nexuscli.cli.subcommand_script](#)), [45](#)
[cmd_delete\(\)](#) (in module [nexuscli.cli.root_commands](#)), [43](#)
[cmd_delete\(\)](#) (in module [nexuscli.cli.subcommand_blobstore](#)), [44](#)
[cmd_delete\(\)](#) (in module [nexuscli.cli.subcommand_repository](#)), [45](#)
[cmd_delete\(\)](#) (in module [nexuscli.cli.subcommand_script](#)), [45](#)
[cmd_download\(\)](#) (in module [nexuscli.cli.root_commands](#)), [43](#)
[cmd_list\(\)](#) (in module [nexuscli.cli.root_commands](#)), [43](#)
[cmd_list\(\)](#) (in module [nexuscli.cli.subcommand_blobstore](#)), [44](#)
[cmd_list\(\)](#) (in module [nexuscli.cli.subcommand_cleanup_policy](#)), [44](#)
[cmd_list\(\)](#) (in module [nexuscli.cli.subcommand_repository](#)), [45](#)
[cmd_list\(\)](#) (in module [nexuscli.cli.subcommand_script](#)), [45](#)
[cmd_list\(\)](#) (in module [nexuscli.cli.subcommand_task](#)), [45](#)
[cmd_login\(\)](#) (in module [nexuscli.cli.root_commands](#)), [43](#)
[cmd_run\(\)](#) (in module [nexuscli.cli.subcommand_script](#)), [45](#)
[cmd_run\(\)](#) (in module [nexuscli.cli.subcommand_task](#)), [45](#)
[cmd_show\(\)](#) (in module [nexuscli.cli.subcommand_blobstore](#)), [44](#)
[cmd_show\(\)](#) (in module [nexuscli.cli.subcommand_cleanup_policy](#)), [44](#)
[cmd_show\(\)](#) (in module [nexuscli.cli.subcommand_repository](#)), [45](#)
[cmd_show\(\)](#) (in module [nexuscli.cli.subcommand_task](#)), [45](#)
[cmd_stop\(\)](#) (in module [nexuscli.cli.subcommand_task](#)), [45](#)
[cmd_upload\(\)](#) (in module [nexuscli.cli.root_commands](#)), [43](#)
[collection\(\)](#) ([nexuscli.api.security.realms.collection.RealmCollection](#) property), [40](#)
[CONFIG_ERROR](#) ([nexuscli.exception.CliReturnCode](#) attribute), [46](#)
[config_file\(\)](#) ([nexuscli.nexus_config.NexusConfig](#) property), [49](#)
[config_file\(\)](#) ([nexuscli.NexusConfig](#) property), [54](#)
[config_path\(\)](#) ([nexuscli.nexus_config.NexusConfig](#) property), [49](#)
[config_path\(\)](#) ([nexuscli.NexusConfig](#) property), [54](#)
[ConfigError](#), [47](#)
[configuration\(\)](#) ([nexuscli.api.base_model.BaseModel](#) property), [42](#)
[configuration\(\)](#) ([nexuscli.api.repository.base_models.group_repository.GroupRepository](#) property), [27](#)
[configuration\(\)](#) ([nexuscli.api.repository.base_models.hosted_repository.HostedRepository](#) property), [28](#)
[configuration\(\)](#) ([nexuscli.api.repository.base_models.proxy_repository.ProxyRepository](#) property), [28](#)
[configuration\(\)](#) ([nexuscli.api.repository.base_models.repository.Repository](#) property), [30](#)
[configuration\(\)](#) ([nexuscli.api.repository.recipes.apr.AprHostedRepository](#) property), [32](#)
[configuration\(\)](#) ([nexuscli.api.repository.recipes.apr.AprProxyRepository](#) property), [32](#)
[configuration\(\)](#) ([nexuscli.api.repository.recipes.docker.DockerProxyRepository](#) property), [33](#)
[CONNECTION_ERROR](#) ([nexuscli.exception.CliReturnCode](#) attribute), [46](#)
[create\(\)](#) ([nexuscli.api.blobstore.collection.BlobstoreCollection](#) method), [23](#)
[create\(\)](#) ([nexuscli.api.repository.collection.RepositoryCollection](#) method), [36](#)
[create\(\)](#) ([nexuscli.api.script.model.ScriptCollection](#) method), [38](#)
[create_if_missing\(\)](#) ([nexuscli.api.script.model.ScriptCollection](#) method), [38](#)
[create_or_update\(\)](#)

(nexuscli.api.cleanup_policy.collection.CleanupPolicyCollection), 25
 create_scripts() (nexuscli.nexus_client.NexusClient method), 48
 create_scripts() (nexuscli.NexusClient method), 55
D
 delete() (nexuscli.api.blobstore.collection.BlobstoreCollection method), 23
 delete() (nexuscli.api.repository.base_models.repository.Repository method), 30
 delete() (nexuscli.api.repository.collection.RepositoryCollection method), 37
 delete() (nexuscli.api.script.model.ScriptCollection method), 39
 DockerGroupRepository (class in nexuscli.api.repository.recipes.docker), 33
 DockerHostedRepository (class in nexuscli.api.repository.recipes.docker), 33
 DockerProxyRepository (class in nexuscli.api.repository.recipes.docker), 33
 download() (nexuscli.api.repository.base_models.repository.Repository method), 30
 DOWNLOAD_ERROR (nexuscli.exception.CliReturnCode attribute), 46
 download_file() (nexuscli.api.repository.base_models.repository.Repository method), 31
 DownloadError, 47
 DST
 nexus3-download command line option, 6
 nexus3-upload command line option, 22
 dump() (nexuscli.nexus_config.NexusConfig method), 49
 dump() (nexuscli.NexusConfig method), 54
 dump_env() (nexuscli.nexus_config.NexusConfig method), 50
 dump_env() (nexuscli.NexusConfig method), 54
E
 ENCRYPTION_TYPES (nexuscli.api.blobstore.model.Blobstore attribute), 24
 ensure_exists() (in module nexuscli.nexus_util), 51
 ensure_known() (in module nexuscli.api.validations), 43
 exists() (nexuscli.api.script.model.ScriptCollection method), 39
 exit_code (nexuscli.exception.ConfigError attribute), 47
 exit_code (nexuscli.exception.DownloadError attribute), 47
 (nexuscli.exception.FeatureNotImplemented attribute), 47
 (nexuscli.exception.NexusClientAPIError attribute), 47
 (nexuscli.exception.NexusClientBaseError attribute), 47
 (nexuscli.exception.NexusClientCapabilityUnsupported attribute), 47
 (nexuscli.exception.NexusClientConnectionError attribute), 47
 (nexuscli.exception.NexusClientCreateCleanupPolicyError attribute), 47
 (nexuscli.exception.NexusClientCreateRepositoryError attribute), 47
 (nexuscli.exception.NexusClientInvalidCleanupPolicy attribute), 48
 (nexuscli.exception.NexusClientInvalidCredentials attribute), 48
 (nexuscli.exception.NexusClientInvalidRepository attribute), 48
 (nexuscli.exception.NotFound attribute), 48
 (nexuscli.exception.TaskDisabled attribute), 48
F
 FeatureNotImplemented, 47
 nexus3-script-create command line option, 18
 filtered_list_gen() (in module nexuscli.nexus_util), 51
G
 get() (nexuscli.api.script.model.ScriptCollection method), 39
 get() (nexuscli.nexus_http.NexusHttp method), 50
 get_by_name() (nexuscli.api.blobstore.collection.BlobstoreCollection method), 23
 get_by_name() (nexuscli.api.cleanup_policy.collection.CleanupPolicyCollection method), 25
 get_by_name() (nexuscli.api.repository.collection.RepositoryCollection method), 37
 get_classes_by_type() (in module nexuscli.api.repository.collection), 37
 get_client() (in module nexuscli.cli.util), 45
 get_command() (nexuscli.cli.util.AliasedGroup method), 45
 get_dst_path_and_file() (in module nexuscli.nexus_util), 52
 get_files() (in module nexuscli.api.repository.base_models.util), 32
 get_raw_by_name() (nexuscli.api.repository.collection.RepositoryCollection method), 37

method), 37
get_repository_class() (in module *nexuscli.api.repository.collection*), 37
get_repository_classes() (in module *nexuscli.api.repository.collection*), 38
get_supported_recipes() (in module *nexuscli.api.repository.collection*), 38
groovy_script() (in module *nexuscli.nexus_util*), 52
GROOVY_SCRIPT_NAME (in module *nexuscli.nexus_util*), 52
GroupRepository (class in *nexuscli.api.repository.base_models.group_repository*), 27

H

has_same_hash() (in module *nexuscli.nexus_util*), 52
HostedRepository (class in *nexuscli.api.repository.base_models.hosted_repository*), 27

I

INDEX_TYPES (in module *nexuscli.api.repository.recipes.docker.DockerProxyRepository*), 33
INVALID_CREDENTIALS (in module *nexuscli.exception.CliReturnCode*), 46
INVALID_SUBCOMMAND (in module *nexuscli.exception.CliReturnCode*), 46

L

list() (in module *nexuscli.api.base_collection.BaseCollection*), 41
list() (in module *nexuscli.api.repository.base_models.repository.Repository*), 31
list_raw() (in module *nexuscli.api.repository.base_models.repository.Repository*), 31
load() (in module *nexuscli.nexus_config.NexusConfig*), 50
load() (in module *nexuscli.NexusConfig*), 54

M

mapped_commands() (in module *nexuscli.cli.util*), 46
MavenGroupRepository (class in *nexuscli.api.repository.recipes.maven*), 33
MavenHostedRepository (class in *nexuscli.api.repository.recipes.maven*), 33
MavenProxyRepository (class in *nexuscli.api.repository.recipes.maven*), 33
move_to_key() (in module *nexuscli.cli.util*), 46

N

NAME
nexus3-blobstore-create-file command line option, 2
nexus3-blobstore-create-s3 command line option, 3
nexus3-blobstore-delete command line option, 4
nexus3-blobstore-show command line option, 4
nexus3-cleanup-policy-create command line option, 5
nexus3-repository-create-group-recipe command line option, 8
nexus3-repository-create-hosted-apt command line option, 9
nexus3-repository-create-hosted-docker command line option, 9
nexus3-repository-create-hosted-maven command line option, 10
nexus3-repository-create-hosted-recipe command line option, 11
nexus3-repository-create-hosted-yum command line option, 11
nexus3-repository-create-proxy-apt command line option, 13
nexus3-repository-create-proxy-docker command line option, 14
nexus3-repository-create-proxy-maven command line option, 15
nexus3-repository-create-proxy-recipe command line option, 16
nexus3-repository-create-proxy-yum command line option, 17
nexus3-script-create command line option, 18
nexus3-script-delete command line option, 19
nexus3-script-run command line option, 19
new() (in module *nexuscli.api.repository.collection.RepositoryCollection*), 37
nexus3 command line option
--version, 2
nexus3-blobstore-create-file command line option
--quota-limit <quota_limit>, 2
--quota-type <quota_type>, 2
-l, 2
-q, 2
NAME, 2
PATH, 2
nexus3-blobstore-create-s3 command line option

```

-E, 3
--bucket-expiration
    <bucket_expiration>, 3
--bucket-name <bucket_name>, 3
--bucket-prefix <bucket_prefix>, 3
--bucket-region <bucket_region>, 3
--encryption-key <encryption_key>, 3
--encryption-type
    <encryption_type>, 3
--key-id <key_id>, 3
--quota-limit <quota_limit>, 3
--quota-type <quota_type>, 3
--role <role>, 3
--secret-key <secret_key>, 3
--session-token <session_token>, 3
-e, 3
-k, 3
-l, 3
-n, 3
-q, 3
-r, 3
-s, 3
NAME, 3
nexus3-blobstore-delete command line
    option
    --yes, 4
    NAME, 4
nexus3-blobstore-list command line
    option
    --json, 4
    --no-json, 4
nexus3-blobstore-show command line
    option
    NAME, 4
nexus3-cleanup-policy-create command
    line option
    --downloaded <downloaded>, 5
    --format <format>, 5
    --notes <notes>, 5
    --regex <regex>, 5
    --updated <updated>, 5
    NAME, 5
nexus3-cleanup-policy-show command
    line option
    POLICY_NAME, 5
nexus3-delete command line option
    REPOSITORY_PATH, 6
nexus3-download command line option
    --cache, 6
    --flatten, 6
    --no-cache, 6
    --no-flatten, 6
    DST, 6
    SRC, 6
nexus3-list command line option
    REPOSITORY_PATH, 7
nexus3-login command line option
    -U, 7
    --no-x509_verify, 7
    --password <password>, 7
    --url <url>, 7
    --username <username>, 7
    --x509_verify, 7
    -p, 7
    -u, 7
nexus3-repository-create-group-recipe
    command line option
    --blob-store-name
        <blob_store_name>, 8
    --member-names <member_names>, 8
    --no-strict-content, 8
    --strict-content, 8
    -m, 8
    NAME, 8
nexus3-repository-create-hosted-apt
    command line option
    --blob-store-name
        <blob_store_name>, 8
    --cleanup-policy <cleanup_policy>, 8
    --distribution <distribution>, 8
    --gpg-keypair <gpg_keypair>, 8
    --no-strict-content, 8
    --passphrase <passphrase>, 9
    --strict-content, 8
    --write-policy <write_policy>, 8
    NAME, 9
nexus3-repository-create-hosted-docker
    command line option
    --blob-store-name
        <blob_store_name>, 9
    --cleanup-policy <cleanup_policy>, 9
    --force-basic-auth, 9
    --http-port <http_port>, 9
    --https-port <https_port>, 9
    --no-force-basic-auth, 9
    --no-strict-content, 9
    --no-v1-enabled, 9
    --strict-content, 9
    --v1-enabled, 9
    --write-policy <write_policy>, 9
    NAME, 9
nexus3-repository-create-hosted-maven
    command line option
    --blob-store-name
        <blob_store_name>, 10
    --cleanup-policy <cleanup_policy>,
        10
    --layout-policy <layout_policy>, 10

```

```
--no-strict-content, 10
--strict-content, 10
--version-policy <version_policy>,
    10
--write-policy <write_policy>, 10
NAME, 10
nexus3-repository-create-hosted-recipe
    command line option
--blob-store-name
    <blob_store_name>, 10
--cleanup-policy <cleanup_policy>,
    10
--no-strict-content, 10
--strict-content, 10
--write-policy <write_policy>, 10
NAME, 11
nexus3-repository-create-hosted-yum
    command line option
--blob-store-name
    <blob_store_name>, 11
--cleanup-policy <cleanup_policy>,
    11
--depth <depth>, 11
--no-strict-content, 11
--strict-content, 11
--write-policy <write_policy>, 11
NAME, 11
nexus3-repository-create-proxy-apt
    command line option
--auto-block, 12
--blob-store-name
    <blob_store_name>, 12
--cleanup-policy <cleanup_policy>,
    12
--content-max-age
    <content_max_age>, 12
--distribution <distribution>, 12
--flat, 12
--health-check, 12
--metadata-max-age
    <metadata_max_age>, 12
--negative-cache, 12
--negative-cache-ttl
    <negative_cache_ttl>, 12
--no-auto-block, 12
--no-flat, 12
--no-health-check, 12
--no-negative-cache, 12
--no-strict-content, 12
--remote-auth-type
    <remote_auth_type>, 12
--remote-password
    <remote_password>, 12
--remote-username
    <remote_username>, 12
--strict-content, 12
NAME, 13
REMOTE_URL, 13
nexus3-repository-create-proxy-docker
    command line option
--auto-block, 13
--blob-store-name
    <blob_store_name>, 13
--cleanup-policy <cleanup_policy>,
    13
--content-max-age
    <content_max_age>, 13
--force-basic-auth, 13
--health-check, 13
--http-port <http_port>, 13
--https-port <https_port>, 14
--index-type <index_type>, 14
--index-url <index_url>, 14
--metadata-max-age
    <metadata_max_age>, 13
--negative-cache, 13
--negative-cache-ttl
    <negative_cache_ttl>, 13
--no-auto-block, 13
--no-force-basic-auth, 13
--no-health-check, 13
--no-negative-cache, 13
--no-strict-content, 13
--no-trust-store, 14
--no-v1-enabled, 13
--remote-auth-type
    <remote_auth_type>, 13
--remote-password
    <remote_password>, 13
--remote-username
    <remote_username>, 13
--strict-content, 13
--trust-store, 14
--v1-enabled, 13
NAME, 14
REMOTE_URL, 14
nexus3-repository-create-proxy-maven
    command line option
--auto-block, 14
--blob-store-name
    <blob_store_name>, 14
--cleanup-policy <cleanup_policy>,
    14
--content-max-age
    <content_max_age>, 14
--health-check, 15
--layout-policy <layout_policy>, 15
```



```

--metadata-max-age
    <metadata_max_age>, 14
--negative-cache, 14
--negative-cache-ttl
    <negative_cache_ttl>, 14
--no-auto-block, 14
--no-health-check, 15
--no-negative-cache, 14
--no-strict-content, 14
--remote-auth-type
    <remote_auth_type>, 14
--remote-password
    <remote_password>, 15
--remote-username
    <remote_username>, 15
--strict-content, 14
--version-policy <version_policy>,
    15
NAME, 15
REMOTE_URL, 15
nexus3-repository-create-proxy-recipe
    command line option
--auto-block, 15
--blob-store-name
    <blob_store_name>, 15
--cleanup-policy <cleanup_policy>,
    15
--content-max-age
    <content_max_age>, 15
--health-check, 16
--metadata-max-age
    <metadata_max_age>, 16
--negative-cache, 15
--negative-cache-ttl
    <negative_cache_ttl>, 15
--no-auto-block, 15
--no-health-check, 16
--no-negative-cache, 15
--no-strict-content, 15
--remote-auth-type
    <remote_auth_type>, 16
--remote-password
    <remote_password>, 16
--remote-username
    <remote_username>, 16
--strict-content, 15
NAME, 16
REMOTE_URL, 16
nexus3-repository-create-proxy-yum
    command line option
--auto-block, 16
--blob-store-name
    <blob_store_name>, 16
--cleanup-policy <cleanup_policy>,
    16
--content-max-age
    <content_max_age>, 16
--health-check, 17
--metadata-max-age
    <metadata_max_age>, 16
--negative-cache, 16
--negative-cache-ttl
    <negative_cache_ttl>, 16
--no-auto-block, 16
--no-health-check, 17
--no-negative-cache, 16
--no-strict-content, 16
--remote-auth-type
    <remote_auth_type>, 17
--remote-password
    <remote_password>, 17
--remote-username
    <remote_username>, 17
--strict-content, 16
NAME, 17
REMOTE_URL, 17
nexus3-repository-delete command line
    option
--yes, 17
REPOSITORY_NAME, 17
nexus3-repository-show command line
    option
REPOSITORY_NAME, 18
nexus3-script-create command line
    option
--script-type <script_type>, 18
FILE, 18
NAME, 18
nexus3-script-delete command line
    option
NAME, 19
nexus3-script-run command line option
--script-arguments
    <script_arguments>, 19
-a, 19
NAME, 19
nexus3-security-realm-activate command
    line option
REALM_ID, 20
nexus3-security-realm-available
    command line option
--json, 20
--no-json, 20
nexus3-task-list command line option
--json, 20
--no-json, 20
nexus3-task-run command line option
TASK_ID, 21

```

nexus3-task-show command line option
TASK_ID, 21

nexus3-task-stop command line option
TASK_ID, 21

nexus3-upload command line option
--flatten, 22
--no-flatten, 22
--no-recurse, 22
--recurse, 22
DST, 22
SRC, 22

nexuscli (module), 54

nexuscli.api (module), 43

nexuscli.api.base_collection (module), 41

nexuscli.api.base_model (module), 42

nexuscli.api.blobstore (module), 25

nexuscli.api.blobstore.collection (module), 23

nexuscli.api.blobstore.model (module), 24

nexuscli.api.cleanup_policy (module), 26

nexuscli.api.cleanup_policy.collection (module), 25

nexuscli.api.cleanup_policy.model (module), 26

nexuscli.api.repository (module), 38

nexuscli.api.repository.base_models (module), 32

nexuscli.api.repository.base_models.base_repository (module), 26

nexuscli.api.repository.base_models.group_repository (module), 27

nexuscli.api.repository.base_models.host_repository (module), 27

nexuscli.api.repository.base_models.proxy_repository (module), 28

nexuscli.api.repository.base_models.repository (module), 29

nexuscli.api.repository.base_models.util (module), 32

nexuscli.api.repository.collection (module), 36

nexuscli.api.repository.model (module), 38

nexuscli.api.repository.recipes (module), 36

nexuscli.api.repository.recipes.apt (module), 32

nexuscli.api.repository.recipes.bower (module), 33

nexuscli.api.repository.recipes.docker (module), 33

nexuscli.api.repository.recipes.maven (module), 33

nexuscli.api.repository.recipes.npm (module), 34

nexuscli.api.repository.recipes.nuget (module), 34

nexuscli.api.repository.recipes.pypi (module), 35

nexuscli.api.repository.recipes.raw (module), 35

nexuscli.api.repository.recipes.rubygems (module), 36

nexuscli.api.repository.recipes.validations (module), 36

nexuscli.api.repository.recipes.yum (module), 36

nexuscli.api.script (module), 39

nexuscli.api.script.model (module), 38

nexuscli.api.security (module), 40

nexuscli.api.security.realms (module), 40

nexuscli.api.security.realms.collection (module), 39

nexuscli.api.security.realms.model (module), 40

nexuscli.api.task (module), 41

nexuscli.api.task.collection (module), 40

nexuscli.api.task.model (module), 41

nexuscli.api.util (module), 42

nexuscli.api.validations (module), 43

nexuscli.cli (module), 46

nexuscli.cli.blobstore_options (module), 43

nexuscli.cli.constants (module), 43

nexuscli.cli.repository_options (module), 43

nexuscli.cli.root_commands (module), 43

nexuscli.cli.subcommand_blobstore (module), 44

nexuscli.cli.subcommand_cleanup_policy (module), 44

nexuscli.cli.subcommand_realm (module), 44

nexuscli.cli.subcommand_repository (module), 45

nexuscli.cli.subcommand_script (module), 45

nexuscli.cli.subcommand_task (module), 45

nexuscli.cli.util (module), 45

nexuscli.exception (module), 46

nexuscli.nexus_client (module), 48

nexuscli.nexus_config (module), 49

nexuscli.nexus_http (module), 50

nexuscli.nexus_util (module), 51

NexusClient (class in nexuscli), 55

NexusClient (class in nexuscli.nexus_client), 48

NexusClientAPIError, 47

NexusClientBaseError, 47

NexusClientCapabilityUnsupported, 47

NexusClientConnectionError, 47

NexusClientCreateCleanupPolicyError, 47
 NexusClientCreateRepositoryError, 47
 NexusClientInvalidCleanupPolicy, 47
 NexusClientInvalidCredentials, 48
 NexusClientInvalidRepository, 48
 NexusClientInvalidRepositoryPath, 48
 NexusConfig (class in *nexuscli*), 54
 NexusConfig (class in *nexuscli.nexus_config*), 49
 NexusHttp (class in *nexuscli.nexus_http*), 50
 NO_FILES (*nexuscli.exception.CliReturnCode* attribute), 46
 NOT_FOUND (*nexuscli.exception.CliReturnCode* attribute), 46
 NOT_IMPLEMENTED (*nexuscli.exception.CliReturnCode* attribute), 46
 NotFound, 48
 NpmGroupRepository (class *nexuscli.api.repository.recipes.npm*), 34
 NpmHostedRepository (class *nexuscli.api.repository.recipes.npm*), 34
 NpmProxyRepository (class *nexuscli.api.repository.recipes.npm*), 34
 NugetGroupRepository (class *nexuscli.api.repository.recipes.nuget*), 34
 NugetHostedRepository (class *nexuscli.api.repository.recipes.nuget*), 34
 NugetProxyRepository (class *nexuscli.api.repository.recipes.nuget*), 34

P

PATH
 nexus3-blobstore-create-file
 command line option, 2
 POLICY_NAME
 nexus3-cleanup-policy-show command
 line option, 5
 POLICY_NOT_FOUND (*nexuscli.exception.CliReturnCode* attribute), 46
 pop_repository() (in module *nexuscli.nexus_util*), 52
 print_as_table() (in module *nexuscli.cli.util*), 46
 ProxyRepository (class in *nexuscli.api.repository.base_models.proxy_repository*), 28
 PypiGroupRepository (class in *nexuscli.api.repository.recipes.pypi*), 35
 PypiHostedRepository (class in *nexuscli.api.repository.recipes.pypi*), 35
 PypiProxyRepository (class in *nexuscli.api.repository.recipes.pypi*), 35

Q

quota_status() (*nexuscli.api.blobstore.collection.BlobstoreCollection* method), 23

QUOTA_TYPES (*nexuscli.api.blobstore.model.Blobstore* attribute), 24

R

raw_active() (*nexuscli.api.security.realms.collection.RealmCollection* method), 40
 raw_list() (*nexuscli.api.base_collection.BaseCollection* method), 41
 raw_list() (*nexuscli.api.blobstore.collection.BlobstoreCollection* method), 23
 raw_list() (*nexuscli.api.cleanup_policy.collection.CleanupPolicyCollection* method), 25
 raw_list() (*nexuscli.api.repository.collection.RepositoryCollection* method), 37
 raw_list() (*nexuscli.api.script.model.ScriptCollection* method), 39
 in raw_list() (*nexuscli.api.security.realms.collection.RealmCollection* method), 40
 in raw_list() (*nexuscli.api.task.collection.TaskCollection* method), 40
 in RawGroupRepository (class in *nexuscli.api.repository.recipes.raw*), 35
 in RawHostedRepository (class in *nexuscli.api.repository.recipes.raw*), 35
 in RawProxyRepository (class in *nexuscli.api.repository.recipes.raw*), 35
 in Realm (class in *nexuscli.api.security.realms.model*), 40
 REALM_ID
 nexus3-security-realm-activate
 command line option, 20
 RealmCollection (class in *nexuscli.api.security.realms.collection*), 39
 RECIPE_NAME (*nexuscli.api.repository.base_models.base_repository.BaseRepository* attribute), 27
 recipe_name() (*nexuscli.api.repository.base_models.base_repository.BaseRepository* property), 27
 REMOTE_PATH_SEPARATOR (in module *nexuscli.nexus_util*), 51
 remote_path_to_local() (in module *nexuscli.nexus_util*), 52
 REMOTE_URL
 nexus3-repository-create-proxy-apt
 command line option, 13
 nexus3-repository-create-proxy-docker
 command line option, 14
 nexus3-repository-create-proxy-maven
 command line option, 15
 nexus3-repository-create-proxy-recipe
 command line option, 16
 nexus3-repository-create-proxy-yum
 command line option, 17
 rename_keys() (in module *nexuscli.cli.util*), 46
 repositories() (*nexuscli.nexus_client.NexusClient* property), 48

repositories() (*nexuscli.NexusClient* property), 55
Repository (class in *nexuscli.api.repository.base_models.repository*), 29
REPOSITORY_NAME
 nexus3-repository-delete command line option, 17
 nexus3-repository-show command line option, 18
REPOSITORY_NOT_FOUND
 (*nexuscli.exception.CliReturnCode* attribute), 46
REPOSITORY_PATH
 nexus3-delete command line option, 6
 nexus3-list command line option, 7
RepositoryCollection (class in *nexuscli.api.repository.collection*), 36
request() (*nexuscli.nexus_http.NexusHttp* method), 50
reset() (*nexuscli.api.base_collection.BaseCollection* method), 41
reset() (*nexuscli.api.security.realms.collection.RealmCollection* method), 40
rest_url() (*nexuscli.nexus_http.NexusHttp* property), 50
RubygemsGroupRepository (class in *nexuscli.api.repository.recipes.rubygems*), 36
RubygemsHostedRepository (class in *nexuscli.api.repository.recipes.rubygems*), 36
RubygemsProxyRepository (class in *nexuscli.api.repository.recipes.rubygems*), 36
run() (*nexuscli.api.script.model.ScriptCollection* method), 39
run() (*nexuscli.api.task.collection.TaskCollection* method), 41
run_script() (*nexuscli.api.base_collection.BaseCollection* method), 41

S
Script (class in *nexuscli.api.script.model*), 38
script_dependencies() (*nexuscli.api.base_collection.BaseCollection* method), 42
script_dependencies() (*nexuscli.api.cleanup_policy.collection.CleanupPolicyCollection* method), 25
script_dependencies() (*nexuscli.api.repository.collection.RepositoryCollection* method), 37
script_for_version() (in module *nexuscli.api.util*), 42
ScriptCollection (class in *nexuscli.api.script.model*), 38
scripts() (*nexuscli.nexus_client.NexusClient* property), 49
scripts() (*nexuscli.NexusClient* property), 55
security_realms() (*nexuscli.nexus_client.NexusClient* property), 49
security_realms() (*nexuscli.NexusClient* property), 55
server_version() (*nexuscli.nexus_http.NexusHttp* property), 50
service_url() (*nexuscli.nexus_http.NexusHttp* property), 51
set_health_check() (*nexuscli.api.repository.collection.RepositoryCollection* method), 37
show() (*nexuscli.api.task.collection.TaskCollection* method), 41
soft_quota() (*nexuscli.api.blobstore.model.Blobstore* property), 24
system_component_path() (in module *nexuscli.nexus_util*), 53
SRC
 nexus3-download command line option, 6
 nexus3-upload command line option, 22
stop() (*nexuscli.api.task.collection.TaskCollection* method), 41
SUBCOMMAND_ERROR (*nexuscli.exception.CliReturnCode* attribute), 46
SUCCESS (*nexuscli.exception.CliReturnCode* attribute), 46

T
Task (class in *nexuscli.api.task.model*), 41
TASK_DISABLED (*nexuscli.exception.CliReturnCode* attribute), 47
TASK_ID
 nexus3-task-run command line option, 21
 nexus3-task-show command line option, 21
 nexus3-task-stop command line option, 21
TaskCollection (class in *nexuscli.api.task.collection*), 40
TaskDisabled, 48
tasks() (*nexuscli.nexus_client.NexusClient* property), 49
tasks() (*nexuscli.NexusClient* property), 55
to_dict() (*nexuscli.nexus_config.NexusConfig* property), 50

`to_dict()` (*nexuscli.NexusConfig* property), 54
`TYPE` (*nexuscli.api.repository.base_models.base_repository.BaseRepository* attribute), 27
`TYPE` (*nexuscli.api.repository.base_models.group_repository.GroupRepository* attribute), 27
`TYPE` (*nexuscli.api.repository.base_models.hosted_repository.XHostedRepository* attribute), 27
`TYPE` (*nexuscli.api.repository.base_models.proxy_repository.ProxyRepository* attribute), 28
`TYPE` (*nexuscli.api.repository.base_models.repository.Repository* attribute), 29
`type()` (*nexuscli.api.blobstore.model.Blobstore* property), 24
`TYPES` (*nexuscli.api.blobstore.model.Blobstore* attribute), 24
`UNKNOWN_ERROR` (*nexuscli.exception.CliReturnCode* attribute), 47
`upcase_values()` (in module *nexuscli.cli.util*), 46
`update()` (*nexuscli.api.blobstore.model.Blobstore* method), 24
`upload()` (*nexuscli.api.repository.base_models.proxy_repository.ProxyRepository* method), 28
`upload()` (*nexuscli.api.repository.base_models.repository.Repository* method), 31
`upload_directory()`
 (*nexuscli.api.repository.base_models.proxy_repository.ProxyRepository* method), 29
`upload_directory()`
 (*nexuscli.api.repository.base_models.repository.Repository* method), 31
`upload_file()` (*nexuscli.api.repository.base_models.repository.Repository* method), 31
`upload_file()` (*nexuscli.api.repository.recipes.npm.NpmHostedRepository* method), 34
`upload_file()` (*nexuscli.api.repository.recipes.pypi.PypiHostedRepository* method), 35
`upload_file()` (*nexuscli.api.repository.recipes.raw.RawHostedRepository* method), 35
`upload_file()` (*nexuscli.api.repository.recipes.yum.YumHostedRepository* method), 36
`url()` (*nexuscli.nexus_config.NexusConfig* property), 50
`url()` (*nexuscli.NexusConfig* property), 55

U

`validate_response()` (in module *nexuscli.api.util*), 42
`validate_strings()` (in module *nexuscli.nexus_util*), 53

W

`with_min_version()` (in module *nexuscli.api.util*), 43