towerlib Documentation

Release 3.14.0

Costas Tyfoxylos

Sep 13, 2023

Contents

1 towerlib **3**
- 1.1 Development Workflow . . . 3
- 1.2 Important Information . . . 4
- 1.3 Project Features . . . 4

2 Installation **5**

3 Usage **7**

4 Contributing **11**
- 4.1 Submit Feedback . . . 11

5 towerlib **13**
- 5.1 towerlib package . . . 13

6 Credits **103**
- 6.1 Development Lead . . . 103
- 6.2 Contributors . . . 103

7 History **105**

8 0.1.0 (25-05-2018) **107**

9 0.2.0 (27-07-2018) **109**

10 0.3.0 (01-08-2018) **111**

11 1.0.0 (27-09-2018) **113**

12 2.0.0 (16-10-2018) **115**

13 2.0.1 (25-10-2018) **117**

14 2.0.2 (25-10-2018) **119**

15 2.0.3 (29-11-2018) **121**

16 2.1.0 (29-11-2018) **123**

17 2.2.0 (03-12-2018) 125

18 2.3.0 (05-12-2018) 127

19 2.3.1 (03-01-2019) 129

20 2.3.2 (09-01-2019) 131

21 2.3.3 (07-03-2019) 133

22 2.3.4 (04-04-2019) 135

23 2.3.5 (25-07-2019) 137

24 2.3.6 (31-07-2019) 139

25 3.0.0 (18-10-2019) 141

26 3.0.1 (18-10-2019) 143

27 3.1.0 (18-10-2019) 145

28 3.2.0 (18-10-2019) 147

29 3.2.1 (01-11-2019) 149

30 3.2.2 (08-11-2019) 151

31 3.2.3 (06-12-2019) 153

32 3.2.4 (12-12-2019) 155

33 3.2.5 (19-12-2019) 157

34 3.2.6 (19-12-2019) 159

35 3.2.7 (14-03-2020) 161

36 3.2.8 (14-03-2020) 163

37 3.2.9 (29-05-2020) 165

38 3.2.10 (09-06-2020) 167

39 3.3.0 (06-07-2020) 169

40 3.4.0 (09-10-2020) 171

41 3.4.1 (16-10-2020) 173

42 3.4.2 (06-11-2020) 175

43 3.5.0 (02-12-2020) 177

44 3.6.0 (06-01-2021) 179

45 3.6.1 (06-01-2021) 181

46 3.6.2 (04-02-2021) 183

47 3.7.0 (09-03-2021) 185

48 3.7.1 (26-04-2021) 187

49 3.7.2 (26-04-2021) 189

50 3.7.3 (02-06-2021) 191

51 3.8.0 (01-07-2021) 193

52 3.9.0 (01-07-2021) 195

53 3.9.1 (10-09-2021) 197

54 3.9.2 (10-09-2021) 199

55 3.9.3 (11-07-2022) 201

56 3.9.4 (01-09-2022) 203

57 3.10.0 (04-10-2022) 205

58 3.11.0 (11-11-2022) 207

59 3.12.0 (10-03-2023) 209

60 3.13.0 (25-04-2023) 211

61 3.13.1 (05-05-2023) 213

62 3.14.0 (13-09-2023) 215

63 Indices and tables 217

Python Module Index 219

Index 221

Contents:

CHAPTER 1

towerlib

A python library to interface with ansible tower's (awx) api.

- Documentation: https://towerlib.readthedocs.org/en/latest

1.1 Development Workflow

The workflow supports the following steps

- lint
- test
- build
- document
- upload
- graph

These actions are supported out of the box by the corresponding scripts under _CI/scripts directory with sane defaults based on best practices. Sourcing setup_aliases.ps1 for windows powershell or setup_aliases.sh in bash on Mac or Linux will provide with handy aliases for the shell of all those commands prepended with an underscore.

The bootstrap script creates a .venv directory inside the project directory hosting the virtual environment. It uses pipenv for that. It is called by all other scripts before they do anything. So one could simple start by calling _lint and that would set up everything before it tried to actually lint the project

Once the code is ready to be delivered the _tag script should be called accepting one of three arguments, patch, minor, major following the semantic versioning scheme. So for the initial delivery one would call

```
$ _tag –minor
```

which would bump the version of the project to 0.1.0 tag it in git and do a push and also ask for the change and automagically update HISTORY.rst with the version and the change provided.

So the full workflow after git is initialized is:

- repeat as necessary (of course it could be test - code - lint :)) * code * lint * test
- commit and push
- develop more through the code-lint-test cycle
- tag (with the appropriate argument)
- build
- upload (if you want to host your package in pypi)
- document (of course this could be run at any point)

1.2 Important Information

This template is based on pipenv. In order to be compatible with requirements.txt so the actual created package can be used by any part of the existing python ecosystem some hacks were needed. So when building a package out of this **do not** simple call

```
$ python setup.py sdist bdist_egg
```

as this will produce an unusable artifact with files missing. Instead use the provided build and upload scripts that create all the necessary files in the artifact.

1.3 Project Features

- Can get, create and delete all entities in a tower deployment.

1.3.1 Todo

This code is MVP and needs a lot of improvements. Some of them are, optimize all queries, implement searching and filtering and implement smart caching of entities.

CHAPTER 2

Installation

At the command line:

```
$ pip install towerlib
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv towerlib
$ pip install towerlib
```

Or, if you are using pipenv:

```
$ pipenv install towerlib
```

CHAPTER 3

Usage

To develop on towerlib:

```
# The following commands require pipenv as a dependency

# To lint the project
_CI/scripts/lint.py

# To execute the testing
_CI/scripts/test.py

# To create a graph of the package and dependency tree
_CI/scripts/graph.py

# To build a package of the project under the directory "dist/"
_CI/scripts/build.py

# To see the package version
_CI/scipts/tag.py

# To bump semantic versioning [--major|--minor|--patch]
_CI/scipts/tag.py --major|--minor|--patch

# To upload the project to a pypi repo if user and password are properly provided
_CI/scripts/upload.py

# To build the documentation of the project
_CI/scripts/document.py
```

To use towerlib in a project:

```
from towerlib import Tower

# using http (pick one)
tower = Tower('ansible.domain.com', 'username', 'password')
```

(continues on next page)

(continued from previous page)

```
# using https with a valid cert (pick one)
tower = Tower('ansible.domain.com', 'username', 'password', secure=True)

# using https with an invalid cert (pick one)
tower = Tower('ansible.domain.com', 'username', 'password', secure=True, ssl_
↪verify=False)

# using https with a custom certificate authority (pick one)
tower = Tower('ansible.domain.com', 'username', 'password', secure=True, ssl_verify='/
↪etc/ssl/certs/example.com.ca.crt')

# using https with an OAUTH2 token (pick one)
tower = Tower('ansible.domain.com', None, None, secure=True, token="token-data-here")

# access hosts
for host in tower.hosts:
    print(host.name)

# access groups
for group in tower.groups:
    print(group.name)

# access inventories
for inventory in tower.inventories:
    print(inventory.name)

# access organizations
for organization in tower.organizations:
    print(organization.name)

# access users
for user in tower.users:
    print(user.username)

# access projects
for project in tower.projects:
    print(project.name)

# access teams
for team in tower.teams:
    print(team.name)

# access jobs
for job in tower.jobs:
    print(job.name)

# all of the above entities also support a dictionary of filters which are provided
# to the api, reducing query time. For filterable terms, see
# https://docs.ansible.com/ansible-tower/latest/html/towerapi/filtering.html

# failed jobs
for job in tower.jobs.filter({"status":"failed"}):
    print(job.name)
# successful jobs
for job in tower.jobs.filter({"status": "successful"}):
    print(job.name)
```

(continues on next page)

(continued from previous page)

```
# all the above entities support creation and deletion either from the core tower␣
↪object
# or from their respective parent object.
# eg: create a host
tower.create_host_in_inventory('inventory_name','host_name', 'host description',
↪'variable_json')

# or
inventory = tower.get_inventory_by_name('inventory_name')
inventory.create_host('host_name', 'host description', 'variable_json')
```

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

4.1 Submit Feedback

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.

4.1.1 Get Started!

Ready to contribute? Here's how to set up *towerlib* for local development.

1. Clone your fork locally:

```
$ git clone https://github.com/costastf/towerlib.git
```

2. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your clone for local development:

```
$ mkvirtualenv towerlib
$ cd towerlib/
$ python setup.py develop
```

3. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

4. Commit your changes and push your branch to the server:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

5. Submit a merge request

CHAPTER 5

towerlib

5.1 towerlib package

5.1.1 Subpackages

towerlib.entities package

Submodules

towerlib.entities.core module

Main code for miscellaneous.

towerlib.entities.core.**Cluster**
alias of towerlib.entities.core.InstanceGroups

class towerlib.entities.core.**ClusterInstance**(*tower_instance*, *name*, *hearbeat*)
Bases: *towerlib.entities.core.DateParserMixin*

Models the instance of a node as part of the cluster.

capacity
The capacity of the node.

consumed_capacity
The consumed capacity.

created_at
The date and time the entity was created in tower.

Returns The datetime object of the date and time of the creation of the object. None: If there is no entry for the creation.

Return type datetime

heartbeat
Datetime object of when the last heartbeat was recorded.

hostname
The hostname of the node.

id
The id of the node.

jobs_running
The number of running jobs.

modified_at
The date and time the entity was modified in tower.

Returns The datetime object of the date and time of the modification of the object. None: If there is no entry for the modification.

Return type datetime

percent_capacity_remaining
The percentage of remaining capacity.

uuid
The uuid of the node.

version
The version of tower on the node.

class towerlib.entities.core.**Config**(*eula*, *license_info*, *analytics_status*, *version*, *project_base_dir*, *time_zone*, *ansible_version*, *project_local_paths*)

Bases: tuple

analytics_status
Alias for field number 2

ansible_version
Alias for field number 6

eula
Alias for field number 0

license_info
Alias for field number 1

project_base_dir
Alias for field number 4

project_local_paths
Alias for field number 7

time_zone
Alias for field number 5

version
Alias for field number 3

class towerlib.entities.core.**DateParserMixin**

Bases: object

Implements a string to datetime parsing to be inherited by all needed objects.

class towerlib.entities.core.**Entity**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.DateParserMixin*

The basic object that holds common responses across all entities.

api_url
The api url of the object.

Returns The relative url of the representation of the object in tower.

Return type string

created_at
The date and time the entity was created in tower.

Returns The datetime object of the date and time of the creation of the object. None: If there is no entry for the creation.

Return type datetime

delete()
Deletes the entity from tower.

Returns True on success, False otherwise.

Return type bool

id
The id of the object.

Returns The number of the internal id of the object in tower.

Return type int

modified_at
The date and time the entity was modified in tower.

Returns The datetime object of the date and time of the modification of the object. None: If there is no entry for the modification.

Return type datetime

type
The type of the object.

Returns The name of the type of the object in tower.

Return type string

url
The url of the object.

Returns The full url of the representation of the object in tower.

Return type string

class towerlib.entities.core.**EntityManager**(*tower_instance*, *entity_object*, *primary_match_field*, *entity_name=None*, *url=None*)
Bases: object

Manages entities by making them act like iterables but also implements contains and other useful stuff.

filter(*params*)
Implements filtering based on the filtering capabilities of tower.

Parameters **params** – Dictionary of filters to be passed to the api.

Returns Generator of the objects retrieved based on the filtering.

https://docs.ansible.com/ansible-tower/latest/html/towerapi/filtering.html.

class towerlib.entities.core.**Label**(*id: int*, *name: str*)

Bases: object

Models a label.

class towerlib.entities.core.**LicenseFeatures**(*surveys*, *multiple_organizations*, *workflows*, *system_tracking*, *enterprise_auth*, *rebranding*, *activity_streams*, *ldap*, *ha*)

Bases: tuple

activity_streams
Alias for field number 6

enterprise_auth
Alias for field number 4

ha
Alias for field number 8

ldap
Alias for field number 7

multiple_organizations
Alias for field number 1

rebranding
Alias for field number 5

surveys
Alias for field number 0

system_tracking
Alias for field number 3

workflows
Alias for field number 2

class towerlib.entities.core.**LicenseInfo**(*subscription_name*, *valid_key*, *features*, *date_expired*, *available_instances*, *hostname*, *free_instances*, *instance_count*, *time_remaining*, *compliant*, *grace_period_remaining*, *trial*, *company_name*, *date_warning*, *license_type*, *license_key*, *license_date*, *deployment_id*, *current_instances*)

Bases: tuple

available_instances
Alias for field number 4

company_name
Alias for field number 12

compliant
Alias for field number 9

current_instances
Alias for field number 18

date_expired
Alias for field number 3

date_warning
Alias for field number 13

deployment_id
Alias for field number 17

features
Alias for field number 2

free_instances
Alias for field number 6

grace_period_remaining
Alias for field number 10

hostname
Alias for field number 5

instance_count
Alias for field number 7

license_date
Alias for field number 16

license_key
Alias for field number 15

license_type
Alias for field number 14

subscription_name
Alias for field number 0

time_remaining
Alias for field number 8

trial
Alias for field number 11

valid_key
Alias for field number 1

towerlib.entities.core.**validate_characters**(*value*, *alpha=True*, *numbers=True*, *extra_chars=None*)
Validates the string groups of a value.

towerlib.entities.core.**validate_json**(*value*)
Validates that the provided value is a valid json.

towerlib.entities.core.**validate_max_length**(*value*, *max_length*)
Validates the maximum length of a value.

towerlib.entities.core.**validate_range**(*value*, *start*, *stop*)
Validates that a value is within a range.

towerlib.entities.credential module

Main code for credentials.

class towerlib.entities.credential.**AmazonWebServicesCredential**(*tower_instance*, *data*)
Bases: *towerlib.entities.credential.GenericCredential*

Models the Amazon Web Services credential.

access_key
The access_key that is set in the credential.

Returns The access_key that is set in the credential.

Return type basestring

secret_key
The secret_key that is set in the credential.

Returns The secret_key that is set in the credential.

Return type basestring

class towerlib.entities.credential.**Credential**
Bases: object

Credential factory to handle the different credential types returned.

class towerlib.entities.credential.**CredentialType**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the credential_type entity of ansible tower.

description
The description of the credential type.

Returns The description of the credential type.

Return type string

injectors
The injectors of the credential type.

Returns A structure of the credential type supported injectors.

Return type dictionary

input_sources
The input sources of the credential.

Returns A structure of the credential input sources.

Return type dictionary

inputs
The inputs of the credential type.

Returns A structure of the credential type supported inputs.

Return type dictionary

kind
The kind of the credential type.

Accepted values are : (u'scm', u'ssh', u'vault', u'net', u'cloud', u'insights').

Returns The kind of the credential type.

Return type string

managed_by_tower
Flag indicating whether the credential is internal to tower.

Returns True if credential is internal to tower, False if it is user created.

Return type bool

name
The name of the credential type.

Returns The name of the credential type.

Return type string

class towerlib.entities.credential.**GenericCredential**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the credential entity of ansible tower.

created_by
The user that created the credential.

Returns The user that created the credential.

Return type *User*

credential_type
The type of the credential.

Returns The type of the credential.

Return type *CredentialType*

description
The description of the credential.

Returns The description of the credential.

Return type string

host
The host of the credential.

Returns The host structure of the credential.

Return type dictionary

inputs
The inputs of the credential.

Returns A structure of the credential supported inputs.

Return type dictionary

modified_by
The person that modified the credential last.

Returns The user that modified the credential in tower last.

Return type *User*

name
The name of the credential.

Returns The name of the credential.

Return type string

object_roles
The object roles.

Returns EntityManager of the object roles supported.

Return type *EntityManager*

organization
The organization the credential is part of.

Returns The organization the credential is part of.

Return type *Organization*

owner_teams
The owner teams of the credential.

Returns EntityManager of the teams that are owners.

Return type *EntityManager*

owner_users
The owners of the credential.

Returns EntityManager of the users that are owners.

Return type *EntityManager*

project
The project of the credential.

Returns The project structure of the credential.

Return type dictionary

class towerlib.entities.credential.**HashicorpVaultCredential**(*tower_instance*, *data*)
Bases: *towerlib.entities.credential.GenericCredential*

Models the hashicorp vault credential.

ca_host_verify
The ca host verify setting that is set in the credential.

Returns The vault address that is set in the credential.

Return type basestring

token
The token that is set in the credential.

Returns The token that is set in the credential.

Return type basestring

vault_address
The vault address that is set in the credential.

Returns The vault address that is set in the credential.

Return type basestring

class towerlib.entities.credential.**MachineCredential**(*tower_instance*, *data*)
Bases: *towerlib.entities.credential.GenericCredential*

Models the machine credential.

password
The password that is set in the credential.

Returns The password that is set in the credential.

Return type basestring

username
The username that is set in the credential.

Returns The username that is set in the credential.

Return type basestring

towerlib.entities.group module

Main code for group.

class towerlib.entities.group.**Group**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the group entity of ansible tower.

add_host_by_name(*name*)
Add a host to the group by name.

Parameters **name** – The name of the host to add to the group.

Returns True on success, False otherwise.

Return type bool

Raises InvalidHost – The host provided as argument does not exist.

associate_group_by_name(*name*)
Associate a group to the group by name.

Parameters **name** – The name of the group to associate with the group.

Returns True on success, False otherwise.

Return type bool

Raises InvalidGroup – The group provided as argument does not exist.

created_by
The user that created the group.

Returns The user that created the group.

Return type *User*

description
The description of the group.

Returns The description of the group.

Return type string

disassociate_group_by_name(*name*)
Disassociate a group from the group.

Parameters **name** – The name of the group to disassociate.

Returns True on success, False otherwise.

Return type bool

Raises InvalidGroup – The group provided as argument does not exist.

groups
The associated groups of the group.

Returns EntityManager of the groups of the group.

Return type *EntityManager*

groups_with_active_failures_count
The number of groups with active failures.

Returns The number of groups with active failures.

Return type integer

has_active_failures
A flag on whether the group has active failures.

Returns True if there are active failures, False if not.

Return type bool

has_inventory_sources
A flag of whether there are.

Returns True if set, False otherwise.

Return type bool

hosts
The hosts of the group.

Returns EntityManager of the hosts of the group.

Return type *EntityManager*

hosts_with_active_failures_count
The number of hosts with active failures.

Returns The number of hosts with active failures.

Return type integer

inventory
The inventory that the group is part of.

Returns The inventory that the group is part of.

Return type *Inventory*

name
The name of the group.

Returns The name of the group.

Return type string

remove_host_by_name(*name*)
Removes a host from the group.

Parameters **name** – The name of the host to remove.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidHost` – The host provided as argument does not exist.

total_groups_count
The number of groups.

Returns The number of groups.

Return type integer

total_hosts_count
The total number of hosts in the group.

Returns The number of group hosts.

Return type integer

variables
The variables set on the group.

Returns A string of the variables set on the group usually in yaml format.

Return type string

towerlib.entities.host module

Main code for host.

class towerlib.entities.host.**Host**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the host entity of ansible tower.

all_groups
The groups that the host is directly and indirectly part of.

Returns EntityManager of the groups of the host.

Return type *EntityManager*

ansible_facts
Returns ansible facts gathered about the host in json.

Args: None

Returns Json representation of ansible facts

Return type json

Raises: None

associate_with_groups(*groups*)
Associate the host with the provided groups.

Parameters

- **groups** – The groups to associate the host with.
- **a single group string or a list or tuple of groups.** (*Accepts*) –

Returns True on complete success, False otherwise.

Return type bool

Raises InvalidGroup – The group provided as argument does not exist.

created_by
The user that created the host.

Returns The user that created the host.

Return type *User*

description
The description of the host.

Returns The description of the host.

Return type string

disassociate_with_groups(*groups*)
Disassociate the host from the provided groups.

Parameters

- **groups** – The group name(s) to disassociate the host from.
- **a single group string or a list or tuple of groups.** (*Accepts*) –

Returns True on complete success, False otherwise.

Return type bool

Raises `InvalidGroup` – The group provided as argument does not exist.

enabled
Flag about whether the host is enabled in tower.

Returns True if the host is enabled, False otherwise.

Return type bool

groups
The groups that the host is part of.

Returns EntityManager of the groups of the host.

Return type *EntityManager*

has_active_failures
Flag about whether the host has active failures.

Returns True if the host has active failures, False otherwise.

Return type bool

has_inventory_sources
Flag about whether the host has inventory sources.

Returns True if the host has inventory sources, False otherwise.

Return type bool

insights_system_id
Not sure what this is.

Returns None.

instance_id
Not sure what this is.

Returns

Return type string

inventory
The inventory that the host is part of.

Returns The inventory that the host is part of.

Return type *Inventory*

job_events
The job_events for the host.

Returns All job events of the host

Return type job_events (list)

last_job
The id of the last job.

Returns The id of the last job.

Return type integer

last_job_host_summary
The id of the last job summary.

Returns The id of the last job summary..

Return type integer

modified_by
The person that modified the host last.

Returns The user that modified the host in tower last.

Return type *User*

name
The name of the host.

Returns The name of the host.

Return type string

recent_jobs
The most recent jobs run on the host.

Returns The most recent jobs run on the host.

Return type list if dict

variables
The variables set on the host.

Returns A string of the variables set on the host usually in yaml format.

Return type string

towerlib.entities.instance module

Main code for instances.

class towerlib.entities.instance.**Instance**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the instance entity of ansible tower.

capacity
Not really sure what this is.

Returns

Return type integer

consumed_capacity
Not really sure what this is.

Returns

Return type integer

hostname
The hostname of the instance.

Returns The hostname of the instance.

Return type string

jobs
The jobs of the instance.

Returns EntityManager of the jobs of the instance.

Return type *EntityManager*

jobs_running_count
The number of running jobs.

Returns The number of running jobs.

Return type integer

percent_capacity_remaining
Not really sure what this is.

Returns

Return type integer

uuid
The uuid of the instance.

Returns The uuid of the instance.

Return type string

version
The version of the instance.

Returns The version of the instance.

Return type string

class towerlib.entities.instance.**InstanceGroup**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the instance_group entity of ansible tower.

capacity
Not really sure what this is.

Returns

Return type integer

consumed_capacity
Not really sure what this is.

Returns

Return type integer

controller
Not really sure what this is.

Returns None.

instances
The instances of the instance group.

Returns The instances of the instance group.

Return type list of Instances

instances_count
The number of instances.

Returns The number of instances.

Return type integer

jobs_running_count
The number of running jobs.

Returns The number of running jobs.

Return type integer

name
The name of the instance group.

Returns The name of the instance group.

Return type string

percent_capacity_remaining
Not really sure what this is.

Returns

Return type integer

towerlib.entities.inventory module

Main code for inventory.

class towerlib.entities.inventory.**Inventory**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the inventory entity of ansible tower.

create_group(*name*, *description*, *variables='{}'*)
Creates a group.

Parameters

- **name** – The name of the group to create.
- **description** – The description of the group.
- **variables** – A json with the variables that will be set on the created group.

Returns The created group is successful, None otherwise.

Return type *Group*

Raises InvalidVariables – The variables provided as argument is not valid json.

create_host(*name*, *description*, *variables='{}'*)
Creates a host.

Parameters

- **name** – The name of the host to create.
- **description** – The description of the host.
- **variables** – A json with the variables that will be set on the created host.

Returns The created host is successful, None otherwise.

Return type *Host*

Raises `InvalidVariables` – The variables provided as argument is not valid json.

create_source(*name*, *description*, *source='scm'*, *source_path=''*, *source_script=None*, *source_vars=''*, *credential=''*, *credential_type=''*, *source_regions=''*, *instance_filters=''*, *group_by=''*, *overwrite=True*, *overwrite_vars=True*, *timeout=0*, *verbosity=1*, *update_on_launch=True*, *update_cache_timeout=0*, *source_project=''*, *update_on_project_update=False*)
Creates a source.

Parameters

- **()** (*update_on_project_update*) –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –

Returns bool

create_source_with_credential_id(*name*, *description*, *credential_id*, *source='scm'*, *source_path=''*, *source_script=None*, *source_vars=''*, *source_regions=''*, *instance_filters=''*, *group_by=''*, *overwrite=True*, *overwrite_vars=True*, *timeout=0*, *verbosity=1*, *update_on_launch=True*, *update_cache_timeout=0*, *source_project=''*, *update_on_project_update=False*)

Creates Source with credential id.

Parameters

- **()** (*update_on_project_update*) –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –
- **()** –

Returns bool

created_by

The user that created the inventory.

Returns The user that created the inventory.

Return type *User*

delete_group(*name*)

Deletes the group.

Parameters **name** – The name of the group to delete.

Returns True on success, False otherwise.

Return type bool

Raises InvalidGroup – The group provided as argument does not exist.

delete_host(*name*)

Deletes the host.

Parameters **name** – The name of the host to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidHost` – The host provided as argument does not exist.

description
The description of the inventory.

Returns The description of the inventory.

Return type string

get_group_by_name(*name*)
Retrieves the group.

Parameters **name** – The name of the group to retrieve.

Returns returns a group if found else None.

Return type group (*Group*)

Raises `InvalidGroup` – The group provided as argument does not exist.

get_host_by_name(*name*)
Retrieves a host.

Parameters **name** – The name of the host to retrieve.

Returns returns a host if found else None.

Return type host (*Host*)

Raises `InvalidHost` – The host provided as argument does not exist.

groups
The groups of the inventory.

Returns The groups of the inventory.

Return type list of Group

groups_with_active_failures_count
The number of groups with active failures.

Returns The number of groups with active failures.

Return type integer

has_active_failures
A flag on whether the inventory has active failures.

Returns True if there are active failures, False if not.

Return type bool

has_inventory_sources
A flag of whether there are.

Returns True if set, False otherwise.

Return type bool

host_filter
Not sure what this does.

Returns The host filter.

Return type string

hosts
The hosts of the inventory.

Returns The hosts of the inventory.

Return type list of Host

hosts_with_active_failures_count
The number of hosts with active failures.

Returns The number of hosts with active failures.

Return type integer

insights_credential
Not sure what this is.

Returns None.

inventory_sources_with_failures_count
The number of sources with failures.

Returns The number of sources with failures.

Return type integer

kind
The kind of inventory.

Returns The kind of inventory.

Return type string

name
The name of the inventory.

Returns The name of the inventory.

Return type string

object_role_names
The names of the object roles.

Returns A list of strings for the object_roles.

Return type list

object_roles
The object roles.

Returns EntityManager of the object roles supported.

Return type *EntityManager*

organization
The organization the inventory is part of.

Returns The organization the inventory is part of.

Return type *Organization*

pending_deletion
Whether the invertory is pending deletion.

Returns True if it is, False otherwise.

Return type bool

sources
The inventory_sources of the inventory.

Returns The inventory_sources of the inventory.

Return type list of Host

total_groups_count
The number of groups.

Returns The number of groups.

Return type integer

total_hosts_count
The total number of hosts in the inventory.

Returns The number of inventory hosts.

Return type integer

total_inventory_sources_count
The number of sources.

Returns The number of sources.

Return type integer

variables
The variables set on the inventory.

Returns A string of the variables set on the inventory usually in yaml format.

Return type string

towerlib.entities.inventory_script module

Main code for inventory_script.

class towerlib.entities.inventory_script.**InventoryScript**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the inventory script entity of ansible tower.

description
The description of the inventory script.

Returns The description of the inventory script.

Return type string

name
The name of the inventory script.

Returns The name of the inventory script.

Return type string

script
The script of the inventory script.

Returns The script of the inventory script.

Return type string

towerlib.entities.inventory_source module

Main code for inventory_source.

class towerlib.entities.inventory_source.**InventorySource**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the inventory source entity of ansible tower.

description
The description of the inventory source.

Returns The description of the inventory source.

Return type string

instance_filters
The instance_filters of the inventory source.

Returns The source path of the inventory source.

Return type string

inventory
The inventory of the inventory source.

Returns The source path of the inventory source.

Return type string

name
The name of the inventory source.

Returns The name of the inventory source.

Return type string

overwrite
The overwrite of the inventory source.

Returns The source path of the inventory source.

Return type string

overwrite_vars
The overwrite_vars of the inventory source.

Returns The source path of the inventory source.

Return type string

source
The source of the inventory source.

Returns The source of the inventory source.

Return type string

source_path
The source_path of the inventory source.

Returns The source path of the inventory source.

Return type string

source_project
The source_project of the inventory source.

Returns The source path of the inventory source.

Return type string

source_regions
The source_regions of the inventory source.

Returns The source path of the inventory source.

Return type string

source_script
The source_script of the inventory source.

Returns The source path of the inventory source.

Return type string

source_vars
The source_vars of the inventory source.

Returns The source path of the inventory source.

Return type string

timeout
The timeout of the inventory source.

Returns The source path of the inventory source.

Return type string

update_cache_timeout
The update_cache_timeout of the inventory source.

Returns The source path of the inventory source.

Return type string

update_on_launch
The update_on_launch of the inventory source.

Returns The source path of the inventory source.

Return type string

update_on_project_update
The update_on_project_update of the inventory source.

Returns The source path of the inventory source.

Return type string

verbosity
The verbosity of the inventory source.

Returns The source path of the inventory source.

Return type string

towerlib.entities.job module

Main code for jobs.

class towerlib.entities.job.**AdHocCommandJob**(*tower_instance*, *data*)
Bases: *towerlib.entities.job.SystemJob*

Models the project update entity of ansible tower.

become_enabled
Boolean on whether the job has become enabled.

Returns Boolean on whether the job has become enabled.

Return type bool

credential
The credential of the job.

Returns The credential of the job.

Return type *Credential*

diff_mode
Boolean on whether the job has diff mode enabled.

Returns Boolean on whether the job has diff mode enabled.

Return type bool

forks_count
Number of forks supported by the job.

Returns Number of forks supported by the job.

Return type integer

inventory
The inventory of the job.

Returns The inventory of the job.

Return type *Inventory*

module_args
The arguments passed to the module.

Returns The arguments passed to the module.

Return type basestring

module_name
The name of the module executed.

Returns The name of the module executed.

Return type basestring

class towerlib.entities.job.**Job**
Bases: object

Job factory to handle the different job types returned.

class towerlib.entities.job.**JobEvent**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the job event entity of ansible tower.

counter
The counter of the event.

Returns The counter of the event.

Return type integer

created_at
The created date of the event.

Returns The string formatted datetime of the event creation.

Return type date_time (string)

end_line
The end line of the event.

Returns The end line of the event.

Return type integer

event
The name of the host.

Returns The name of the event.

Return type string

event_data
The data of the event.

Returns The data of the event.

Return type basestring

event_display
The display of the event.

Returns The display of the event.

Return type basestring

event_level
The level of the event.

Returns The level of the event.

Return type integer

host
The host of the event.

Returns The host of the event.

Return type basestring

host_name
The hostname of the event.

Returns The hostname of the event.

Return type basestring

is_changed
Whether the event is changed.

Returns Whether the event is changed.

Return type bool

is_failed
Whether the event is failed.

Returns Whether the event is failed.

Return type bool

job
The job this event belongs to.

Returns The job this event belongs to.

Return type *Job*

modified_at
The modified date of the event.

Returns The string formatted datetime of the event's last modification.

Return type basestring

parent
The parent of the event.

Returns The parent of the event.

Return type basestring

parent_uuid
The parent uuid of the event.

Returns The parent uuid of the event.

Return type basestring

play
The play of the event.

Returns The play of the event.

Return type basestring

playbook
The playbook of the event.

Returns The playbook of the event.

Return type basestring

role
The role of the event.

Returns The role of the event.

Return type basestring

start_line
The start line of the event.

Returns The start line of the event.

Return type integer

stdout
The stdout of the event.

Returns The stdout of the event.

Return type basestring

task
The task of the event.

Returns The task of the event.

Return type basestring

uuid
The uuid of the event.

Returns The uuid of the event.

Return type basestring

verbosity
The verbosity of the event.

Returns The verbosity of the event.

Return type integer

class towerlib.entities.job.**JobRun**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the Job entity of ansible tower.

cancel()
Cancels the running or pending job.

Returns True on success, False otherwise.

created_by
The User that created the job.

Returns The user that created the job in tower.

Return type *User*

credentials
The credentials of this job.

Returns The credentials this job has.

Return type Credentials

description
The description of the job.

Returns The description of the job.

Return type string

elapsed_time
The elpsed time of the job.

Returns The seconds elapsed since the start of the job.

Return type elapsed

extra_credentials
The extra credentials of the job.

Returns EntityManager of the extra credentials.

Return type *EntityManager*

extra_vars
The extra_vars of the job.

Returns The extra_vars of the job

Return type extra_vars

finished
The finished datetime of the job.

Returns The datetime of the end of the job.

Return type finished_datetime

host
The host of the credential.

Returns The host structure of the credential.

Return type dictionary

inventory
The inventory this job belongs to.

Returns The inventory this job belongs to.

Return type *Inventory*

job_events
The job events of the runs of this job.

Returns EntityManager of the job events.

Return type *EntityManager*

job_host_summaries
The job summaries of the runs of this job.

Returns EntityManager of the job host summaries.

Return type *EntityManager*

job_template
The job template of this job.

Returns The job template of this job.

Return type *JobTemplate*

job_type
The type of the job.

Returns The type of the job.

Return type string

modified_at
The modification datetime of the job.

Returns The datetime of the modification.

Return type modification_datetime

name
The name of the job.

Returns The name of the job.

Return type string

project
The project this job belongs to.

Returns The project this job belongs to.

Return type *Project*

result_stdout
The result stdout of the job.

Returns The result stdout of the job.

Return type result_stdout

result_traceback
The result traceback of the job.

Returns The result traceback of the job.

Return type result_traceback

status
The status of the job.

Returns The status of the job.

Return type status

stdout
The stdout of the job execution.

Returns The stdout of the job execution.

Return type basestring

summary_fields
The summary fields of the job.

Returns The summary fields of the job.

Return type dict

unified_job_template
The job template this job belongs to.

Returns The job template this job belongs to.

Return type *JobTemplate*

class towerlib.entities.job.**JobSummary**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the Job entity of ansible tower.

failed
Whether the job is failed or not.

Returns Whether the job is failed or not.

Return type bool

failures_count
The number of job failures.

Returns The number of job failures.

Return type integer

host
The host of the job summary.

Returns The host of the job summary.

Return type basestring

host_name
The host name of the job summary.

Returns The host name of the job summary.

Return type basestring

is_changed
Whether the job had been changed.

Returns Whether the job had been changed.

Return type integer

is_dark
Whether the job is dark.

Returns Whether the job is dark.

Return type integer

is_ok
Whether the job is dark.

Returns Whether the job is dark.

Return type integer

processed_count
The number of time the job was processed.

Returns The number of time the job was processed.

Return type integer

skipped_count
The number of time the job was skipped.

Returns The number of time the job was skipped.

Return type integer

summary_fields
The summary_fields of this job summary.

Returns The summary_fields of this job summary.

Return type dict

class towerlib.entities.job.**JobTemplate**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the Job Template entity of ansible tower.

add_credential(*credential*, *credential_type='Machine'*)
Adds credential by name.

Parameters

- **credential** (*str*) – A single string of a credential.

- **credential_type** (*int*) – The type of the credential.

add_credential_by_id(*credential_id*)
Adds credential by id.

Parameters **credential_id** (*int*) – The id of the credential.

add_schedule(*name*, *start_date*, *start_time*, *limit=''*, *description=''*, *time_zone='Europe/Berlin'*, *repeat_frequency='DAILY'*, *interval=1*, *extra_vars: dict = None*)
Adds a schedule to a job template.

Parameters

- **name** (*str*) – A name of the schedule.
- **start_date** (*datetime.date*) – The start date of the schedule
- **start_time** (*datetime.time*) – Start time of the schedule
- **limit** (*str*) – Limit of the schedule
- **description** (*str*) – Description of the schedule
- **time_zone** (*str*) – The time zone assigned to the schedule
- **repeat_frequency** (*str*) – How frequently the job will be run
- **interval** (*int*) – Interval of the Job
- **extra_vars** (*dict*) – Variables of the schedule

allow_simultaneous
Flag about whether to allow simultaneous executions.

Returns True if set to allow simultaneous executions , False otherwise.

Return type bool

ask_credential_on_launch
Flag about whether to ask for credential on launch.

Returns True if set to ask for credential on launch, False otherwise.

Return type bool

ask_diff_mode_on_launch
Flag about whether to ask for diff mode on launch.

Returns True if set to ask for diff mode on launch, False otherwise.

Return type bool

ask_inventory_on_launch
Flag about whether to ask for inventory on launch.

Returns True if set to ask for inventory on launch, False otherwise.

Return type bool

ask_job_type_on_launch
Flag about whether to ask for job type on launch.

Returns True if set to ask for job type on launch, False otherwise.

Return type bool

ask_limit_on_launch
Flag about whether to ask for limit on launch.

Returns True if set to ask for limit on launch, False otherwise.

Return type bool

ask_skip_tags_on_launch
Flag about whether to ask to skip tags on launch.

Returns True if set to ask to skip tags on launch, False otherwise.

Return type bool

ask_tags_on_launch
Flag about whether to ask for tags on launch.

Returns True if set to ask for tags on launch, False otherwise.

Return type bool

ask_variables_on_launch
Flag about whether to ask for variables on launch.

Returns True if set to ask for variables on launch, False otherwise.

Return type bool

ask_verbosity_on_launch
Flag about whether to ask for verbosity on launch.

Returns True if set to ask for verbosity on launch, False otherwise.

Return type bool

become_enabled
Not really sure what this is.

Returns

Return type bool

created_by
The User that created the job template.

Returns The user that created the job template in tower.

Return type *User*

credentials
The credentials that the job template uses.

Returns The credentials that the job template uses.

Return type Credentials

description
The description of the job template.

Returns The description of the job template.

Return type string

diff_mode
Flag about whether the diff mode is enabled.

Returns True if diff mode is enabled, False otherwise.

Return type bool

extra_credentials
The extra_credentials that the job template uses.

Returns EntityManager of the extra credentials.

Return type *EntityManager*

extra_vars
The extra vars of the job template.

Returns The extra vars of the job template.

Return type string

force_handlers
Flag about whether the handlers are forced.

Returns True if the handlers are forced, False otherwise.

Return type bool

forks_count
The number of forks of the job template.

Returns The number of forks of the job template.

Return type string

host_config_key
Not really sure what this is.

Returns

Return type string

inventory
The inventory that the job template is part of.

Returns The inventory that the job template is part of.

Return type *Inventory*

is_last_job_failed
Flag about whether the last job failed.

Returns True if last run job failed, False otherwise.

Return type bool

job_tags
The job tags of the job template.

Returns The job tags of the job template.

Return type string

job_type
The job_type of the job template.

Returns The job_type of the job template.

Return type string

labels
The labels of the job template.

Returns The job template labels (only id and name can be retrieved).

Return type labels (list)

last_job_run_at
Date and time that the last job template job was run.

Returns The datetime of the last job execution.

Return type datetime

last_job_run_id
The id of most recent job run on the template.

Returns The id of the most recent job run on the template

Return type int/None

launch(*extra_vars=None*, *job_tags=None*, *limit=None*, *inventory=None*, *credential=None*, *credentials=None*, *scm_branch=None*, *verbosity=None*, *skip_tags=None*, *job_type=None*, *diff_mode=None*)
Launches the job template.

https://docs.ansible.com/ansible-tower/latest/html/towerapi/launch_jobtemplate.html.

Returns Job object of the running job on success, None otherwise.

Return type *Job*

limit
The limit of the job template.

Returns The limit of the job template.

Return type string

modified_by
The User that modified the job template last.

Returns The user that modified the job template in tower last.

Return type *User*

name
The name of the job template.

Returns The name of the job template.

Return type string

next_job_run_at
Date and time when the next job template job will run.

Returns The datetime of the next job execution.

Return type datetime

object_roles
The object roles.

Returns The object roles supported.

Return type *ObjectRole*

playbook
The playbook of the job template.

Returns The playbook of the job template.

Return type string

project
The project that the job template is part of.

Returns The project that the job template is part of.

Return type *Inventory*

recent_jobs
The most recent jobs on the template.

Returns The most recent jobs run on the template.

Return type list if dict

schedules
The schedules that the job template uses.

Returns EntityManager of the schedules.

Return type *EntityManager*

set_inventory(*organization*, *name*)
Inventory from specified organization applied as a prompt, assuming job template prompts for inventory.

Parameters

- **organization** (*str*) – The organization name the inventory belongs to
- **name** (*str*) – The inventory name to set. To reset the inventory set it to empty string.

Returns None.

skip_tags
The tags to skip for the job template.

Returns The tags to skip for the job template.

Return type string

start_at_task
Not really sure what this is.

Returns

Return type string

status
The status of the job template.

Returns The status of the job template.

Return type string

summary_fields
The summary fields of the job template.

Returns The summary fields of the job template.

Return type dict

survey_enabled
Flag about whether the survey mode is enabled.

Returns True if survey mode is enabled, False otherwise.

Return type bool

survey_spec
Survey specification.

Returns Dict that contains info about the linked survey if exists

Return type dict

timeout
The timeout setting of the job template.

Returns The timeout setting of the job template.

Return type integer

use_fact_cache
Flag about whether the fact cache should be used.

Returns True if the fact cache should be used, False otherwise.

Return type bool

vault_credential
The vault credential of the job template.

Returns The vault credential of the job template.

Return type string

verbosity
The verbosity level of the job template.

Returns The verbosity level of the job template.

Return type string

class towerlib.entities.job.**ProjectUpdateJob**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the project update entity of ansible tower.

credentials
The credentials of the project.

Returns The credentials of the project.

Return type Credentials

description
The description of the project for the update.

Returns The description of the project for the update.

Return type basestring

elapsed
The time elapsed during the run.

Returns The time elapsed during the run.

Return type float

execution_node
The node the job got executed on.

Returns The node the job got executed on.

Return type basestring

finished_at
The date and time the update was finished.

Returns The date and time the update was finished. None: If there is no entry for the finish time.

Return type datetime

is_failed
The status of the update.

Returns The status of the update.

Return type bool

job_args
The arguments passed to the execution of the job.

Returns The arguments passed to the execution of the job.

Return type basestring

job_cwd
The current working directory of the execution of the job.

Returns The current working directory of the execution of the job.

Return type basestring

job_env
The environment of the execution of the job.

Returns The environment of the execution of the job.

Return type dict

job_explanation
The explanation of the job.

Returns The explanation of the job.

Return type basestring

job_type
The type of the job executed on the update.

Returns The type of the job executed on the update.

Return type basestring

launch_type
The type of launching for the update.

Returns The type of launching for the update.

Return type basestring

local_path
The local path of the project for the update.

Returns The local path of the project for the update.

Return type basestring

name
The name of the project for the update.

Returns The name of the project for the update.

Return type basestring

project
The project of the ProjectUpdateJob.

Returns The Project that this project_update belongs to.

Return type *Project*

result_stdout
The stdout of the result of the job.

Returns The stdout of the result of the job.

Return type basestring

result_traceback
The traceback of the result.

Returns The traceback of the result.

Return type basestring

scm_branch
The scm branch of the project for the update.

Returns The scm branch of the project for the update.

Return type basestring

scm_clean
Whether the scm clean flag is set on the project.

Returns Whether the scm clean flag is set on the project.

Return type bool

scm_delete_on_update
Whether the scm delete on update flag is set on the project.

Returns Whether the scm delete on update flag is set on the project.

Return type bool

scm_type
The scm type of the project for the update.

Returns The scm type of the project for the update.

Return type basestring

scm_url
The scm url of the project for the update.

Returns The scm url of the project for the update.

Return type basestring

started_at
The date and time the update was started.

Returns The date and time the update was started. None: If there is no entry for the start time.

Return type datetime

status
The status of the update.

Returns The status of the update.

Return type basestring

stdout
The stdout of the project update.

Returns The stdout of the project update.

Return type basestring

summary_fields
The summary fields of the job.

Returns The summary fields of the job.

Return type dict

timeout
Timeout setting of the project.

Returns Timeout setting of the project.

Return type integer

class towerlib.entities.job.**SystemJob**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the Job entity of ansible tower.

description
The description of the job.

Returns The description of the job.

Return type string

elapsed
The time elapsed during the run.

Returns The time elapsed during the run.

Return type float

execution_node
The node the job got executed on.

Returns The node the job got executed on.

Return type basestring

extra_vars
The extra vars of the job template.

Returns The extra vars of the job template.

Return type string

finished_at
The date and time the job was finished.

Returns:_ datetime: The datetime object of the date and time of the end of the job. None: If there is no entry for the finish.

is_failed
The status of the update.

Returns The status of the update.

Return type bool

job_args
The arguments passed to the execution of the job.

Returns The arguments passed to the execution of the job.

Return type basestring

job_cwd
The current working directory of the execution of the job.

Returns The current working directory of the execution of the job.

Return type basestring

job_env
The environment of the execution of the job.

Returns The environment of the execution of the job.

Return type dict

job_explanation
The explanation of the job.

Returns The explanation of the job.

Return type basestring

job_type
The type of the job.

Returns The type of the job.

Return type string

launch_type
The type of launching for the update.

Returns The type of launching for the update.

Return type basestring

name
The name of the job.

Returns The name of the job.

Return type string

result_stdout
The stdout of the result of the job.

Returns The stdout of the result of the job.

Return type basestring

result_traceback
The traceback of the result.

Returns The traceback of the result.

Return type basestring

started_at
The date and time the job was started.

Returns:_ datetime: The datetime object of the date and time of the start of the job. None: If there is no entry for the start.

status
The status of the update.

Returns The status of the update.

Return type basestring

summary_fields
The summary fields of the job.

Returns The summary fields of the job.

Return type dict

class towerlib.entities.job.**WorkflowJobRun**(*tower_instance*, *data*)
Bases: *towerlib.entities.job.JobRun*

Models the Workflow Job Run entity of ansible tower.

cancel()
Cancels the running or pending job.

Returns True on success, False otherwise.

workflow_nodes
Workflow nodes is resouce belonging to a Workflow job.

Returns This resouce belonging to a Workflow job.

Return type *WorkflowNodes*

class towerlib.entities.job.**WorkflowNodes**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the Workflow nodes entity of ansible tower.

count
Number of workflow nodes.

Returns Number of workflow nodes.

Return type integer

next
The *next* and *previous* fields provides links to additional results if there are more than will fit on a single page.

Returns url of next page.

Return type string

previous
The *next* and *previous* fields provides links to additional results if there are more than will fit on a single page.

Returns url of previous page.

Return type string

results
Results for workflow nodes.

Returns Results for workflow nodes.

Return type list

towerlib.entities.notification module

Main code for notification.

class towerlib.entities.notification.**Notification**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the notifications of Ansible Tower/AWX.

error
The error status for the notification.

Returns The error

Return type string

notification_type
The notification type for which the notification has been sent.

Returns The notification type

Return type string

notifications_sent
How many notifications have been sent.

Returns Total notifications sent

Return type int

recipients
The recipients to whom the notification has been sent.

Returns List of recipients

Return type []string

status
The status of the notification.

Returns The status of the application can be pending,successful,failed

Return type string

subject
The subject of the notification.

Returns The subject of the notification

Return type string

class towerlib.entities.notification.**NotificationEmail**(*data*)
Bases: object

Notification type configuration for email.

host
Host to where we send the email.

Returns Host to where we send the email

Return type string

password
The password to use.

Returns Password

Return type string

port
The port to where to send the email.

Returns Port

Return type int

recipients
Recipient list.

Returns Recipient list

Return type []string

sender
Sender email.

Returns Sender Email

Return type string

timeout
Timeout.

Returns The timeout (defaults to 30)

Return type int

use_ssl
Use SSL for the connection?

Returns Use SSL for the connection

Return type bool

use_tls
Use TLS for the connection?

Returns Use TLS for the connection

Return type bool

username
The username to use.

Returns Username

Return type string

class towerlib.entities.notification.**NotificationGrafana**(*data*)
Bases: object

Notification type configuration for WebHook.

grafana_key
Get the grafana key.

Returns The grafana key

Return type string

grafana_url
The URL to call for the notification.

Returns The URL to call for a notification

Return type string

class towerlib.entities.notification.**NotificationHipChat**(*data*)
Bases: object

Notification type configuration for HipChat.

api_url
//mycompany.hipchat.com).

Returns API Url

Return type string

Type API Url (e.g

Type https

color
Notification color.

Returns Notification color

Return type string

message_from
Label to be shown with notification.

Returns Label to be shown with notification

Return type string

notify
Notify room.

Returns Notify room?

Return type bool

rooms
Destination Rooms.

Returns Destination Rooms

Return type []string

token
The token.

Returns token

Return type string

class towerlib.entities.notification.**NotificationIRC**(*data*)
Bases: object

Notification type configuration for IRC.

nickname
IRC Nick.

Returns The irc nick

Return type string

password
The IRC Server Password.

Returns IRC Server Password

Return type string

port
The IRC Server Port.

Returns IRC Server Port

Return type int

server
IRC Server Address.

Returns The irc server address

Return type string

targets
Destination channels or users.

Returns Destination channels or users

Return type []string

use_ssl
Use SSL for the connection?

Returns Use SSL for the connection

Return type bool

class towerlib.entities.notification.**NotificationMatterMost**(*data*)
Bases: object

Notification type configuration for MatterMost.

mattermost_no_verify_ssl
Do not verify SSL on MatterMost.

Returns Do we verify SSL?

Return type bool

mattermost_url
The URL to call for the notification.

Returns The URL to call for a notification

Return type string

class towerlib.entities.notification.**NotificationPagerDuty**(*data*)
Bases: object

Notification type configuration for pagerduty.

client_name
The client name for the PagerDuty.

Returns The client name for PagerDuty

Return type string

service_key
The service key for the PagerDuty.

Returns The service key for PagerDuty

Return type string

subdomain
Gets the pagerduty subdomain.

Returns Pagerduty subdomain

Return type string

token
The token for the PagerDuty.

Returns The token for PagerDuty

Return type string

class towerlib.entities.notification.**NotificationRocketChat**(*data*)
Bases: object

Notification type configuration for Rocket.Chat.

rocketchat_no_verify_ssl
Do not verify SSL on Rocket.Chat.

Returns Do we verify SSL?

Return type bool

rocketchat_url
The URL to call for the notification.

Returns The URL to call for a notification

Return type string

class towerlib.entities.notification.**NotificationSlack**(*data*)
Bases: object

Notification type configuration for slack.

channels
The channels to where we send the notification to.

Returns List of channels to notify

Return type []string

token
Token required to make an API call.

Returns The token for the API call

Return type string

class towerlib.entities.notification.**NotificationTemplate**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the notification template of Ansible Tower/AWX.

description
The description of the notification template.

Returns The description of the notification template

Return type string

name
The name of the notification template.

Returns The name of the notification template

Return type string

notification_configuration
Gets the notification configuration.

Returns The configuration for the notification

Return type dict

notification_type
Notification type for the template.

Returns Notification type for the template

Return type string

organization
The Organization object that this project is part of.

Returns The Organization object that this project is part of

Return type *Organization*

recent_notifications
The groups configured in Tower/AWX.

Returns The manager object for groups

Return type *EntityManager*

class towerlib.entities.notification.**NotificationTwilio**(*data*)
Bases: object

Notification type configuration for twilio.

account_sid
The account sid.

Returns Account SID

Return type string

account_token
Account Token.

Returns Account Token

Return type string

from_number
Source phone number.

Returns The source phone number

Return type string

to_numbers
Destination SMS numbers.

Returns Destination SMS numbers

Return type []string

class towerlib.entities.notification.**NotificationWebHook**(*data*)
Bases: object

Notification type configuration for WebHook.

disable_ssl_verification
Disable SSL verification.

Returns Do we verify SSL?

Return type bool

url
The URL to call for the notification.

Returns The URL to call for a notification

Return type string

towerlib.entities.organization module

Main code for organization.

class towerlib.entities.organization.**Organization**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the organization entity of ansible tower.

DEFAULT_MEMBER_ROLE = 'Member'

admins_count
The number of administrators of the organization.

Returns The count of the administrators on the organization.

Return type integer

create_inventory(*name*, *description*, *variables='{}'*)
Creates an inventory.

Parameters

- **name** – The name of the inventory to create.
- **description** – The description of the inventory.
- **variables** – A json with the initial variables set on the inventory.

Returns The created Inventory object on success, None otherwise.

Return type *Inventory*

Raises InvalidVariables – The variables provided as argument is not valid json.

create_inventory_script(*name*, *description*, *script*)
Creates a custom inventory script.

Parameters

- **name** – Name of the inventory script.
- **description** – The description of the inventory script.
- **script** – The script of the inventory script.

Returns The created inventory script is successful, None otherwise.

Return type Inventory_script

create_project(*name*, *description*, *credential*, *scm_url*, *local_path=''*, *custom_virtualenv=''*, *scm_branch='master'*, *scm_type='git'*, *scm_clean=True*, *scm_delete_on_update=False*, *scm_update_on_launch=True*, *scm_update_cache_timeout=0*)

Creates a project in the organization.

Parameters

- **name** (*str*) – The name of the project.
- **description** (*str*) – The description of the project.
- **credential** (*str*) – The name of the credential to use for the project or '' if None.
- **scm_url** (*str*) – The url of the scm.
- **local_path** (*str*) – Local path (relative to PROJECTS_ROOT) containing playbooks and files for this project.
- **custom_virtualenv** (*str*) – Local absolute file path containing a custom Python virtualenv to use.
- **scm_branch** (*str*) – The default branch of the scm.
- **scm_type** (*str*) – The type of the scm.
- **scm_clean** (*bool*) – Clean scm or not.
- **scm_delete_on_update** (*bool*) – Delete scm on update.
- **scm_update_on_launch** (*bool*) – Update scm on launch.
- **scm_update_cache_timeout** (*int*) – Scm cache update.

Returns The created project on success, None otherwise.

Return type *Project*

Raises `InvalidCredential` – The credential provided as argument does not exist.

create_team(*name*, *description*)

Creates a team.

Parameters

- **name** – The name of the team to create.
- **description** – The description of the team.

Returns The created Team object on success, None otherwise.

Return type *Team*

created_by

The User that created the organization.

Returns The user that created the organization in tower.

Return type *User*

credentials

The credentials of the organization.

Returns EntityManager of the credentials of the organization.

Return type *EntityManager*

custom_virtualenv
The path of the custom virtual environment.

Returns The path of the custom virtual environment.

Return type string

delete_inventory(*name*)
Deletes an inventory by name.

Parameters **name** – The name of the inventory to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidInventory` – The inventory provided as argument does not exist.

delete_inventory_script(*name*)
Deletes a custom inventory script.

Parameters **name** – The name of the custom inventory script to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidInventoryScript` – The inventory_script provided as argument does not exist.

delete_project(*name*)
Deletes a project by username.

Parameters **name** – The name of the project to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidProject` – The project provided as argument does not exist.

delete_team(*name*)
Deletes a team by name.

Parameters **name** – The name of the team to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidTeam` – The team provided as argument does not exist.

description
The description of the Organization.

Returns The description of the Organization.

Return type string

get_credential_by_id(*id_*)
Retrieves a credential by id.

Parameters **id** – The id of the credential to retrieve.

Returns The credential if a match is found else None.

Return type *Host*

get_credential_by_name(*name*, *credential_type*)
Retrieves credential matching a certain name.

Parameters

- **name** – The name of the credential to retrieve.
- **credential_type** – The type of credential.

Returns A credential if found else none.

Return type *Credential*

Raises `InvalidCredentialType` – The credential type given as a parameter was not found.

get_credential_by_name_with_type_id(*name*, *credential_type_id*)
Retrieves credential matching a certain name and the provided type by id.

Parameters

- **name** (*str*) – The name of the credential to retrieve.
- **credential_type_id** (*int*) – The type of credential.

Returns A credential if found else none.

Return type *Credential*

get_inventory_by_name(*name*)
Retrieves an inventory.

Parameters **name** – The name of the inventory to retrieve.

Returns inventory on success else None.

Return type inventory(*Inventory*)

get_inventory_script_by_name(*name*)
Retrieves an custom inventory script.

Parameters **name** – The name of the custom inventory script to retrieve.

Returns custom inventory script on success else None.

Return type inventory(*Inventory*)

get_project_by_name(*name*)
Retrieves a project.

Parameters **name** – The name of the project to retrieve.

Returns project on success else None.

Return type project (*Project*)

get_team_by_name(*name*)
Retrieves a team.

Parameters **name** – The name of the team to retrieve.

Returns team on success else None.

Return type team (*Team*)

inventories
The inventories of the organization.

Returns EntityManager of the inventories of the organization.

Return type *EntityManager*

inventories_count
The number of inventories of the organization.

Returns The count of the inventories on the organization.

Return type integer

inventory_scripts
The inventory scripts of the organization.

Returns EntityManager of the inventory scripts of the organization.

Return type *EntityManager*

job_templates_count
The number of job templates of the organization.

Returns The count of the job templates on the organization.

Return type integer

modified_by
The User that modified the organization last.

Returns The user that modified the organization in tower last.

Return type *User*

name
The name of the Organization.

Returns The name of the organization.

Return type string

object_role_names
The names of the object roles.

Returns A list of strings for the object_roles.

Return type list

object_roles
The object roles.

Returns EntityManager of the roles supported.

Return type *EntityManager*

projects
The projects of the organization.

Returns EntityManager of the projects.

Return type *EntityManager*

projects_count
The number of projects of the organization.

Returns The count of the projects on the organization.

Return type integer

teams
The teams of the organization.

Returns EntityManager of the teams of the organization.

Return type *EntityManager*

teams_count
The number of teams of the organization.

Returns The count of the teams on the organization.

Return type integer

users
The users of the organization.

Returns EntityManager of the users of the organization.

Return type *EntityManager*

users_count
The number of user of the organization.

Returns The count of the users on the organization.

Return type integer

towerlib.entities.project module

Main code for project.

class towerlib.entities.project.**Project**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the project entity of ansible tower.

created_by
The person that created the project.

Returns The person that created the project.

Return type dict

credential
The Credential object associated with this project.

Returns The Credential object of the project.

Return type *Credential*

custom_virtualenv
The path of the custom virtual environment.

Returns The path of the custom virtual environment.

Return type string

description
The description of the Project.

Returns The description of the Project.

Return type string

is_last_job_failed
A flag of whether the last job run failed.

Returns True if the job failed, False otherwise.

Return type bool

is_last_update_failed
Flag on whether the last update failed or not.

Returns True if last update has failed, False otherwise.

Return type bool

job_templates
The job templates for the project entity.

Returns list of all the job templates object for the project.

Return type job_templates (list)

last_job
The last job run on the project.

Returns The last job executed.

Return type dict

last_job_run
The date and time of the last run job.

Returns The datetime object of when the last job run.

Return type datetime

last_update
The last update of the project.

Returns The last update.

Return type dict

last_updated
The date and time of the last update.

Returns The datetime of the last update, None if not set.

Return type datetime

local_path
The internal local path of the project.

Returns The internal local path of the project.

Return type string

name
The name of the project.

Returns The name of the project.

Return type string

next_job_run
Not sure what this does.

Returns None.

object_role_names
The names of the object roles.

Returns A list of strings for the object_roles.

Return type list

object_roles
The object roles.

Returns EntityManager of the object roles supported.

Return type *EntityManager*

organization
The Organization object that this project is part of.

Returns The Organization object that this project is part of.

Return type *Organization*

playbooks
The playbooks of the project.

Returns A list of the project specified playbooks.

Return type list

project_updates
Get all the job_events for host.

Returns list of all the job events for the host.

Return type list

scm_branch
The branch of the scm used.

Returns The branch of the scm used.

Return type string

scm_clean
A flag to clean the scm or not.

Returns True if set, False otherwise.

Return type bool

scm_delete_on_next_update
A flag on whether to delete the scm on the next update.

Returns True if set, False otherwise.

Return type bool

scm_delete_on_update
A flag to delete the scm on update.

Returns True if set, False otherwise.

Return type bool

scm_revision
The hash of the scm revision.

Returns The hash of the scm revision.

Return type string

scm_type
The type of the scm used.

Returns The type of the scm used.

Return type string

scm_update_cache_timeout
The cache timeout set for the scm.

Returns The cache time out set.

Return type integer

scm_update_on_launch
A flag on whether to update scm on launch.

Returns True if set, False otherwise.

Return type bool

scm_url
The url of the scm used.

Returns The url of the scm used.

Return type string

status
The status of the project.

Returns The status of the project.

Return type string

timeout
The timeout setting of the project.

Returns The timeout setting of the project.

Return type integer

update()
Send an SCM update request to the project.

Returns Response of api request as json on success, None otherwise.

Return type dict

towerlib.entities.role module

Main code for role.

class towerlib.entities.role.**ObjectRole**(*tower_instance*, *data*)
Bases: *towerlib.entities.role.Role*

Models the object role entity of ansible tower.

team
The team that has the object role assigned.

Returns The team that has the object role assigned.

Return type *Team*

class towerlib.entities.role.**Role**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the role entity of ansible tower.

description
The description of the role.

Returns The description of the role.

Return type string

name
The name of the role.

Returns The name of the role.

Return type string

projects
The projects of the team.

Returns EntityManager of the projects.

Return type *EntityManager*

summary_fields
The summary fields of the role.

Returns The summary fields of the role.

Return type *Organization*

teams
The teams that have the role assigned.

Returns EntityManager of the teams.

Return type *EntityManager*

users
The users of the team.

Returns EntityManager of the users.

Return type *EntityManager*

towerlib.entities.schedule module

Main code for Schedule.

class towerlib.entities.schedule.**Schedule**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the schedule entity of ansible tower.

datetime_end
The last occurrence of the schedule occurs before this time, aftewards the schedule expires.

Returns The last occurrence of the schedule occurs before this time, aftewards the schedule expires.

Return type datetime

datetime_start
The first occurrence of the schedule occurs on or after this time.

Returns The first occurrence of the schedule occurs on or after this time.

Return type datetime

description
The description of the schedule.

Returns The description of the schedule.

Return type string

diff_mode
Display mode for the execution the schedule.

Returns Are we displaying diff mode for the run?

Return type boolean

enabled
Enables processing of this schedule.

Returns Enables processing of this schedule.

Return type bool

extra_data
The extra data of the schedule.

Returns The extra data of the schedule.

Return type dict

inventory
The inventory this schedule belongs to.

Returns The inventory this schedule belongs to.

Return type *Inventory*

job_tags
The job tags of the schedule.

Returns The job tags of the schedule.

Return type string

job_type
The job_type of the schedule.

Returns The job_type of the schedule.

Return type string

limit
The limit of the schedule.

Returns The limit of the schedule.

Return type string

name
The name of the schedule.

Returns The name of the schedule.

Return type string

next_run
The next time that the scheduled action will run.

Returns The next time that the scheduled action will run.

Return type datetime

recurrence_rule
A value representing the schedules iCal recurrence rule.

Returns A value representing the schedules iCal recurrence rule.

Return type string

scm_branch
The scm_branch of the schedule.

Returns The scm_branch of the schedule.

Return type string

skip_tags
The tags to skip of the schedule.

Returns The tags to skip of the schedule.

Return type string

timezone
The timezone of the schedule.

Returns The timezone of the schedule.

Return type string

unified_job_template
Unified job template.

Returns Unified job template .

Return type *JobTemplate*

until
Until when does the schedule run?

Returns Until when does the schedule run?

Return type string

verbosity
Verbosity of the run.

Returns Verbosity of the run .

Return type string

towerlib.entities.settings module

Main code for settings.

class towerlib.entities.settings.**Saml**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the saml entity of ansible tower.

callback_url
The saml callback url.

Returns The saml callback url.

Return type string

enabled_idps
The configured IDPS as a dictionary.

Returns The configured IDPS as a dictionary.

Return type string

extra_data
The IDP attributes that are mapped to extra_attributes.

Returns The IDP attributes that are mapped to extra_attributes.

Return type string

metadata_url
The saml metadata url.

Returns The saml metadata url.

Return type string

organization_attributes
The saml callback url.

Returns The saml callback url.

Return type string

organization_information
The organization information url.

Returns The organization information url.

Return type string

organization_map
The mapping to organization admins/users from social auth accounts.

Returns The mapping to organization admins/users from social auth accounts.

Return type string

security_config
The saml security config.

Returns The saml security config.

Return type string

sp_entity_id
The application-defined unique identifier for SAML service provider (SP) configuration.

Returns The application-defined unique identifier for SAML service provider (SP) configuration.

Return type string

sp_extra
The Service Provider configuration setting.

Returns The Service Provider configuration setting.

Return type string

sp_private_key
The private key.

Returns The private key.

Return type string

sp_public_cert
The public certificate.

Returns The public certificate.

Return type string

support_contact
The support contact information.

Returns The support contact information.

Return type string

team_attributes
The translation of user team membership into Tower.

Returns The translation of user team membership into Tower.

Return type string

team_map
The mapping of team members (users) from social auth accounts.

Returns The mapping of team members (users) from social auth accounts.

Return type string

technical_contact
The technical contact information.

Returns The technical contact information.

Return type string

url
The url of the object.

Returns The full url of the representation of the object in tower.

Return type string

class towerlib.entities.settings.**Settings**(*tower_instance*)
Bases: object

Models the settings entity of ansible tower.

saml
The saml settings in tower.

Returns The saml settings in tower.

Return type *Saml*

towerlib.entities.team module

Main code for team.

class towerlib.entities.team.**Team**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the team entity of ansible tower.

add_credential_permission_admin(*credential_name*, *credential_type*)
Adds a credential with admin permissions.

Parameters

- **credential_name** – The name of the credential to add.
- **credential_type** (*str*) – The type of the credential to use

Returns True on success, False otherwise.

add_credential_permission_use(*credential_name*, *credential_type*)
Adds a credential with admin permissions.

Parameters

- **credential_name** – The name of the credential to add.
- **credential_type** (*str*) – The type of the credential to use

Returns True on success, False otherwise.

add_inventory_permission_ad_hoc(*inventory_name*)
Adds an inventory with ad hoc permissions.

Parameters **inventory_name** – The name of the inventory to add.

Returns True on success, False otherwise.

add_inventory_permission_admin(*inventory_name*)
Adds an inventory with admin permissions.

Parameters **inventory_name** – The name of the inventory to add.

Returns True on success, False otherwise.

add_inventory_permission_update(*inventory_name*)
Adds an inventory with update permissions.

Parameters **inventory_name** – The name of the inventory to add.

Returns True on success, False otherwise.

add_inventory_permission_use(*inventory_name*)
Adds an inventory with use permissions.

Parameters **inventory_name** – The name of the inventory to add.

Returns True on success, False otherwise.

add_job_template_permission_admin(*job_template_name*)
Adds a job template with admin permissions.

Parameters **job_template_name** – The name of the job template to add.

Returns True on success, False otherwise.

add_job_template_permission_execute(*job_template_name*)
Adds a job template with execute permissions.

Parameters **job_template_name** – The name of the job template to add.

Returns True on success, False otherwise.

add_organization_role_by_name(*organization_name*, *role_name*)
Adds an organization role to the team.

Parameters

- **organization_name** (*str*) – The name of the organization to search roles for.
- **role_name** (*str*) – The name of the role to add.

Returns True on success, False otherwise.

add_project_permission_admin(*project_name*)
Adds a project with admin permissions.

Parameters **project_name** – The name of the project to add.

Returns True on success, False otherwise.

add_project_permission_update(*project_name*)
Adds a project with update permissions.

Parameters **project_name** – The name of the project to add.

Returns True on success, False otherwise.

add_project_permission_use(*project_name*)
Adds a project with use permissions.

Parameters **project_name** – The name of the project to add.

Returns True on success, False otherwise.

add_user_as_admin(*username*)
Adds a user as an admin of the team.

Parameters **username** – The username of the user to add.

Returns True on success, False otherwise.

add_user_as_member(*username*)
Adds a user as a member of the team.

Parameters **username** – The username of the user to add.

Returns True on success, False otherwise.

credentials
The credentials of the team.

Returns EntityManager of the credentials.

Return type *EntityManager*

description
The description of the team.

Returns The description of the team.

Return type string

get_user_by_username(*username*)
Retrieves a user of the team by its username.

Parameters **username** – The username of the user to retrieve.

Returns user (User) on success, None otherwise.

name
The name of the team.

Returns The name of the team.

Return type string

object_role_names
The names of the object roles.

Returns A list of strings for the object_roles.

Return type list

object_roles
The object roles.

Returns EntityManager of the object roles supported.

Return type *EntityManager*

organization
The Organization of the team.

Returns The organization of the team.

Return type *Organization*

projects
The projects of the team.

Returns EntityManager of the projects.

Return type *EntityManager*

remove_credential_permission_admin(*credential_name*, *credential_type*)
Removes a credential with admin permissions.

Parameters

- **credential_name** – The name of the credential to remove.
- **credential_type** (*str*) – The type of the credential to use

Returns True on success, False otherwise.

remove_credential_permission_use(*credential_name*, *credential_type*)
Removes a credential with use permissions.

Parameters

- **credential_name** – The name of the credential to remove.
- **credential_type** (*str*) – The type of the credential to use

Returns True on success, False otherwise.

remove_inventory_permission_ad_hoc(*inventory_name*)
Removes an inventory with ad hoc permissions.

Parameters **inventory_name** – The name of the inventory to remove.

Returns True on success, False otherwise.

remove_inventory_permission_admin(*inventory_name*)
Removes an inventory with admin permissions.

Parameters **inventory_name** – The name of the inventory to remove.

Returns True on success, False otherwise.

remove_inventory_permission_update(*inventory_name*)
Removes an inventory with update permissions.

Parameters **inventory_name** – The name of the inventory to remove.

Returns True on success, False otherwise.

remove_inventory_permission_use(*inventory_name*)
Removes an inventory with use permissions.

Parameters **inventory_name** – The name of the inventory to remove.

Returns True on success, False otherwise.

remove_job_template_permission_admin(*job_template_name*)
Removes a job template with admin permissions.

Parameters **job_template_name** – The name of the job template to remove.

Returns True on success, False otherwise.

remove_job_template_permission_execute(*job_template_name*)
Removes a job template with execute permissions.

Parameters **job_template_name** – The name of the job template to remove.

Returns True on success, False otherwise.

remove_organization_role_by_name(*organization_name*, *role_name*)
Removes an organization role from the team.

Parameters

- **organization_name** (*str*) – The name of the organization to search roles for.
- **role_name** (*str*) – The name of the role to add.

Returns True on success, False otherwise.

remove_project_permission_admin(*project_name*)
Removes a project with admin permissions.

Parameters **project_name** – The name of the project to remove.

Returns True on success, False otherwise.

remove_project_permission_update(*project_name*)
Removes a project with update permissions.

Parameters **project_name** – The name of the project to remove.

Returns True on success, False otherwise.

remove_project_permission_use(*project_name*)
Removes a project with use permissions.

Parameters **project_name** – The name of the project to remove.

Returns True on success, False otherwise.

remove_user_as_admin(*username*)
Removes a user as an admin of the team.

Parameters **username** – The username of the user to remove.

Returns True on success, False otherwise.

remove_user_as_member(*username*)
Removes a user as a member of the team.

Parameters **username** – The username of the user to remove.

Returns True on success, False otherwise.

roles
The roles.

Returns EntityManager of the roles.

Return type *EntityManager*

users
The users of the team.

Returns EntityManager of the users.

Return type *EntityManager*

towerlib.entities.user module

Main code for user.

class towerlib.entities.user.**User**(*tower_instance*, *data*)
Bases: *towerlib.entities.core.Entity*

Models the user entity of ansible tower.

associate_with_organization_role(*organization*, *role*)
Associate a user to an organizational role.

Parameters

- **organization** – The organization that we want to assign to
- **role** – The role we want to assign to the object

Returns If it managed to associate the user to the organization

Return type bool

auth
The authentication setting for the user.

Returns Used authentication methods set for the user.

Return type list

credentials
The credentials that the user has.

Returns EntityManager of the credentials.

Return type *EntityManager*

disassociate_from_organization_role(*organization*, *role*)
Disassociate a user to an organizational role.

Parameters

- **organization** – The organization object that we want to assign to
- **role** – The role we want to assign to the object

Returns If it managed to disassociate the user to the organization

Return type bool

email
The email of the user.

Returns The email of the user.

Return type string

external_account
The external account entry for the user.

Returns The external account entry for the user if it exists. None: If no entry exists.

Return type string

first_name
The first name of the user.

Returns The first name of the user.

Return type string

is_superuser
The superuser status of the user.

Returns True if the user is a superuser, False otherwise.

Return type bool

is_system_auditor
The system auditor status of the user.

Returns True if the user is a system auditor, False otherwise.

Return type bool

last_login
The last time the user logged in to the system.

Returns The datetime object of the date and time of the last login for the user. None: If there is no entry for the last login date.

Return type datetime

last_name
The last name of the user.

Returns The last name of the user.

Return type string

ldap_dn
The ldap dn setting for the user.

Returns The ldap dn entry for the user.

Return type string

organizations
The organizations that the user is part of.

Returns EntityManager of the organizations.

Return type *EntityManager*

password
The password of the user.

Returns The masked password value of the user.

Return type string

projects
The projects that the user is part of.

Returns EntityManager of the projects.

Return type *EntityManager*

roles
The roles that the user has.

Returns EntityManager of the roles.

Return type *EntityManager*

teams
The teams that the user is part of.

Returns EntityManager of the teams.

Return type *EntityManager*

username
The username of the user.

Returns The username of the user.

Return type string

Module contents

towerlib package.

Import all parts from entities here.

5.1.2 Submodules

5.1.3 towerlib.towerlib module

Main code for towerlib.

class towerlib.towerlib.**Tower**(*host*, *username*, *password*, *secure=False*, *ssl_verify=True*, *token=None*, *pool_connections=10*, *pool_maxsize=25*, *timeout=5*)
Bases: object

Models the api of ansible tower.

static add_slash(*url*)
Adds a final slash to a url if there is not any.

associate_groups_with_inventory_host(*organization*, *inventory*, *hostname*, *groups*)
Adds groups to a host.

Parameters

- **organization** – The name of the organization the inventory belongs to.
- **inventory** – The inventory to retrieve the host from.
- **hostname** – The name of the host to add the groups to.
- **groups** – A string of a single group or a list or tuple of group names to add to host.

Returns True on complete success, False otherwise.

Return type bool

Raises `InvalidHost` – The host provided as argument does not exist.

cluster
The cluster status of tower.

Returns The information about the state of the cluster.

Return type Cluster

configuration
The configuration of the tower instance.

Returns The configuration of the tower instance.

Return type *Config*

create_credential_in_organization(*organization*, *name*, *description*, *credential_type*, *user=None*, *team=None*, *inputs_='{}'*)
Creates a credential under an organization.

Parameters

- **organization** – The name of the organization to create a credential under.
- **name** – The name of the credential to create.
- **description** – The description of the credential to create.
- **user** – The username of the user to assign to the credential.
- **team** – The name of the team to assign to the credential.
- **credential_type** – The name of the type of the credential.
- **inputs** – A json with the values to set to the credential according to what is required by its type.

Returns The created credential upon success, None otherwise.

Return type *Credential*

Raises

- `InvalidOrganization` – The organization provided as argument does not exist.
- `InvalidCredentialType` – The credential type provided as argument does not exist.
- `InvalidVariables` – The inputs provided as argument is not valid json.
- `InvalidUser` – The user provided as argument does not exist.
- `InvalidTeam` – The team provided as argument does not exist.

create_credential_in_organization_with_type_id(*organization*, *name*, *description*, *user*, *team*, *credential_type_id*, *inputs_='{}'*)
Creates a credential under an organization.

Parameters

- **organization** (*str*) – The name of the organization to create a credential under.
- **name** (*str*) – The name of the credential to create.
- **description** (*str*) – The description of the credential to create.
- **user** (*str*) – The username of the user to assign to the credential.

- **team** (*str*) – The name of the team to assign to the credential.
- **credential_type_id** (*int*) – The number of the type of the credential.
- **inputs** (*str*) – A json with the values to set to the credential according to what is required by its type.

Returns The created credential upon success, None otherwise.

Return type *Credential*

Raises

- `InvalidOrganization` – The organization provided as argument does not exist.
- `InvalidUser` – The user provided as argument does not exist.
- `InvalidTeam` – The team provided as argument does not exist.
- `InvalidVariables` – The inputs provided as argument is not valid json.

create_credential_type(*name*, *description*, *type_*, *inputs_='{}'*, *injectors='{}'*)

Creates a credential type in tower.

Parameters

- **name** – The name of the credential type.
- **description** – The description of the credential type.
- **type** – The kind of credential type.Valid values (u'scm', u'ssh', u'vault', u'net', u'cloud', u'insights').
- **inputs** (*str*) – A json of the inputs to set to the credential type.
- **injectors** (*str*) – A json of the injectors to set to the credential type.

Returns CredentialType on success, None otherwise.

Raises

- `InvalidCredentialTypeKind` – The credential type kind provided as argument does not exist.
- `InvalidVariables` – The inputs or injectors provided as argument is not valid json.

create_credential_with_credential_type_id(*name: str*, *credential_type_id: int*, *description=''*, *organization_id=None*, *user_id=None*, *team_id=None*, *inputs='{}'*)

Creates a credential using the id of the provided credential type.

Parameters

- **name** (*str*) – The name of the credential
- **credential_type_id** (*int*) – The number of the credential type
- **description** (*str*) – The description of the credential
- **organization_id** (*int*) – The id of the organization
- **user_id** (*int*) – The id of the user
- **team_id** (*int*) – The id of the team
- **inputs** (*str*) – The input to provide to the credential as json in a string format

Returns A credential object on success, false otherwise.

Return type credential (Credential|None)

create_host_in_inventory(*organization*, *inventory*, *name*, *description*, *variables='{}'*)
Creates a host under an inventory.

Parameters

- **organization** – The name of the organization the inventory belongs to.
- **inventory** – The name of the inventory to create the host under.
- **name** – The name of the host.
- **description** – The description of the host.
- **variables** – A json of the variables to be set on the host.

Returns The created host on success, None otherwise.

Return type *Host*

Raises `InvalidInventory` – The inventory provided as argument does not exist.

create_inventory_group(*organization*, *inventory*, *name*, *description*, *variables='{}'*)
Creates a group in an inventory in tower.

Parameters

- **organization** – The organization the inventory belongs to.
- **inventory** – The name of the inventory to create the group in.
- **name** – The name of the group to create.
- **description** (*str*) – The description of the group to create.
- **variables** (*str*) – The Variables of the group in a json string format.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidGroup` – The group provided as argument does not exist.

create_job_template(*name*, *description*, *organization*, *inventory*, *project*, *playbook*, *credential=None*, *credential_type=None*, *instance_groups=None*, *host_config_key=None*, *job_type='run'*, *vault_credential=None*, *forks=0*, *limit=0*, *verbosity=0*, *extra_vars=''*, *job_tags=''*, *force_handlers=False*, *skip_tags=''*, *start_at_task=''*, *timeout=0*, *use_fact_cache=False*, *ask_diff_mode_on_launch=False*, *ask_variables_on_launch=False*, *ask_limit_on_launch=False*, *ask_tags_on_launch=False*, *ask_skip_tags_on_launch=False*, *ask_job_type_on_launch=False*, *ask_verbosity_on_launch=False*, *ask_inventory_on_launch=False*, *ask_credential_on_launch=False*, *survey_enabled=False*, *become_enabled=False*, *diff_mode=False*, *allow_simultaneous=False*)
Creates a job template.

Parameters

- **name** – The name of the job template to create.
- **description** – The description of the job template to create.
- **organization** – The organization the inventory belongs to.
- **inventory** – The inventory to use for the template.

- **project** – The project to use for the template.
- **playbook** – The playbook to run for the template.
- **credential** – The credential to use for the template.
- **credential_type** – The type of the credential to use for the template.
- **instance_groups** – The instance groups to associate to the template.
- **host_config_key** – A host config key.
- **job_type** – The job type. Valid values are 'run' and 'check'.
- **vault_credential** – A vault credential.
- **forks** – The number of parallel or simultaneous processes to use while executing the playbook.
- **limit** – A host pattern to constrain the list of hosts that will be managed or affected by the playbook.
- **verbosity** – The level of output ansible will produce as the playbook executes. Values [0-4].
- **extra_vars** – Pass extra command line variables to the playbook.
- **job_tags** – Tags to identify the template.
- **force_handlers** –
- **skip_tags** – Skip specific parts of a play or task with tags.
- **start_at_task** –
- **timeout** –
- **use_fact_cache** –
- **ask_diff_mode_on_launch** –
- **ask_variables_on_launch** –
- **ask_limit_on_launch** –
- **ask_tags_on_launch** –
- **ask_skip_tags_on_launch** –
- **ask_job_type_on_launch** –
- **ask_verbosity_on_launch** –
- **ask_inventory_on_launch** –
- **ask_credential_on_launch** –
- **survey_enabled** –
- **become_enabled** –
- **diff_mode** –
- **allow_simultaneous** –

Returns The created job template if successful, None otherwise.

Return type *JobTemplate*

Raises

- `InvalidInventory` – The inventory provided as argument does not exist.
- `InvalidProject` – The project provided as argument does not exist.
- `InvalidPlaybook` – The playbook provided as argument does not exist in project.
- `InvalidInstanceGroup` – The instance group provided as argument does not exist.
- `InvalidJobType` – The job type provided as argument does not exist.
- `InvalidVerbosity` – The verbosity provided is not in valid range of 0-4.
- `InvalidCredentialType` – The credential type is invalid.

create_organization(*name*, *description=''*)
Creates an organization in tower.

Parameters

- **name** – The name of the organization to create.
- **description** – The description of the organization to create.

Returns The organization on success, None otherwise.

Return type *Organization*

create_organization_inventory(*organization*, *name*, *description*, *variables='{}'*)
Creates an inventory under an organization.

Parameters

- **organization** – The name of the organization to create the inventory under
- **name** – The name of the inventory
- **description** – The description of the inventory
- **variables** – A json of the variables to be set on the inventory

Returns The created inventory on success, None otherwise

Return type *Inventory*

Raises `InvalidOrganization` – The organization provided as argument does not exist.

create_organization_inventory_script(*organization*, *name*, *description*, *script*)
Creates a custom inventory script.

Parameters

- **organization** – The organization the inventory script is part of.
- **name** – Name of the inventory script.
- **description** – The description of the inventory script.
- **script** – The script of the inventory script.

Returns The created inventory script is successful, None otherwise.

Return type Inventory_script

create_project_in_organization(*organization*, *name*, *description*, *credential*, *scm_url*, *local_path=''*, *custom_virtualenv=''*, *scm_branch='master'*, *scm_type='git'*, *scm_clean=True*, *scm_delete_on_update=False*, *scm_update_on_launch=True*, *scm_update_cache_timeout=0*)

Creates a project in an organization.

Parameters

- **organization** (*str*) – The name of the organization to create the project under.
- **name** (*str*) – The name of the project.
- **description** (*str*) – The description of the project.
- **credential** (*str*) – The name of the credential to use for the project.
- **scm_url** (*str*) – The url of the scm.
- **local_path** (*str*) – Local path (relative to PROJECTS_ROOT) containing playbooks and files for this project.
- **custom_virtualenv** (*str*) – Local absolute file path containing a custom Python virtualenv to use.
- **scm_branch** (*str*) – The default branch of the scm.
- **scm_type** (*str*) – The type of the scm.
- **scm_clean** (*bool*) – Clean scm or not.
- **scm_delete_on_update** (*bool*) – Delete scm on update.
- **scm_update_on_launch** (*bool*) – Update scm on launch.
- **scm_update_cache_timeout** (*int*) – Scm cache update.

Returns The created project on success, None otherwise.

Return type *Project*

Raises InvalidOrganization – The organization provided as argument does not exist.

create_team_in_organization(*organization*, *team_name*, *description=''*)

Creates a team under an organization.

Parameters

- **organization** – The name of the organization to create the team under.
- **team_name** – The name of the team to create.
- **description** – The description of the team to create.

Returns The created team on success, None otherwise.

Return type *Team*

Raises InvalidOrganization – The organization provided as argument does not exist.

create_user(*username*, *password*, *first_name=''*, *last_name=''*, *email=''*, *is_superuser=False*, *is_system_auditor=False*)

Creates a user in AWX/Tower.

Parameters

- **username** – The username to create for the user.

- **password** – The password to set for the user.
- **first_name** – The first name of the user.
- **last_name** – The last name of the user.
- **email** – The email of the user.
- **is_superuser** – Is the user a super user
- **is_system_auditor** – Is the user an auditor

Returns The created User object on success, None otherwise.

Return type *User*

create_user_in_organization(*organization*, *username*, *password*, *first_name*, *last_name*, *email*)

Creates a user in an organization.

Parameters

- **organization** – The name of the organization to create the user under.
- **username** – The user's username.
- **password** – The user's password.
- **first_name** – The user's first name.
- **last_name** – The user's last name.
- **email** – The user's email.

Returns The user on success, None otherwise.

Return type *User*

Raises `InvalidOrganization` – The organization provided as argument does not exist.

credential_types

The credential_types configured in tower.

Returns The manager object for credentials type.

Return type *EntityManager*

credentials

The credentials configured in tower.

Returns The manager object for credentials.

Return type *EntityManager*

custom_credential_types

The custom credential_types configured in tower.

Returns The manager object for external credential types.

Return type *EntityManager*

delete_credential_type(*name*)

Deletes a credential_type from tower.

Parameters **name** – The name of the credential_type to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidCredential` – The credential provided as argument does not exist.

delete_inventory_group(*organization*, *inventory*, *name*)
Deletes a group from tower.

Parameters

- **organization** – The organization the inventory belongs to.
- **inventory** – The name of the inventory to retrieve the group from.
- **name** – The name of the group to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidGroup` – The group provided as argument does not exist.

delete_inventory_host(*organization*, *inventory*, *name*)
Deletes an host from tower.

Parameters

- **organization** – The name of the organization the inventory belongs to.
- **inventory** – The name of the inventory to delete the host from.
- **name** – The name of the host to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidHost` – The host provided as argument does not exist.

delete_job_template(*name*)
Deletes a job template from tower.

Parameters **name** – The name of the job template to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidJobTemplate` – The job template provided as argument does not exist.

delete_organization(*name*)
Deletes an organization from tower.

Parameters **name** – The name of the organization to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidOrganization` – The organization provided as argument does not exist.

delete_organization_credential_by_name(*organization*, *name*, *credential_type*)
Deletes a credential from an organization.

Parameters

- **organization** – The organization that owns the credential.
- **name** – The name of the credential(s) to delete.
- **credential_type** – The type of the credential.

Returns True on success, False otherwise.

Return type bool

Raises

- `InvalidCredentialType` – The CredentialType given was not found.
- `InvalidOrganization` – The Organization given was not found.
- `InvalidCredential` – The credential was not found.

delete_organization_credential_by_name_with_type_id(*organization*, *name*, *credential_type_id*)

Deletes a credential from an organization.

Parameters

- **organization** (*str*) – The organization that owns the credential.
- **name** (*str*) – The name of the credential(s) to delete.
- **credential_type_id** (*int*) – The type of the credential.

Returns True on success, False otherwise.

Return type bool

Raises

- `InvalidOrganization` – The Organization given was not found.
- `InvalidCredential` – The credential was not found.

delete_organization_inventory(*organization*, *name*)

Deletes an inventory from tower.

Parameters

- **organization** – The organization the inventory is a member of.
- **name** – The name of the inventory to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidInventory` – The inventory provided as argument does not exist.

delete_organization_inventory_script(*organization*, *name*)

Deletes an custom inventory script from tower.

Parameters

- **organization** – The organization the custom inventory script is a member of.
- **name** – The name of the custom inventory script to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidInventory` – The custom inventory script provided as argument does not exist.

delete_organization_project(*organization*, *name*)

Deletes a project from tower.

Parameters

- **organization** – The organization the project belongs to.

- **name** – The name of the project to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidProject` – The project provided as argument does not exist.

delete_team_in_organization(*organization*, *name*)
Deletes a team from tower.

Parameters

- **organization** – The name of the organization the team belongs to.
- **name** – The name of the team to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidTeam` – The team provided as argument does not exist.

delete_user(*username*)
Deletes a user by username.

Parameters **username** – The username of the user to delete.

Returns True on success, False otherwise.

Return type bool

Raises `InvalidUser` – The username provided as argument does not exist.

disassociate_groups_from_inventory_host(*organization*, *inventory*, *hostname*, *groups*)
Removes groups from a host.

Parameters

- **organization** – The name of the organization the inventory belongs to.
- **inventory** – The inventory which contains the host to affect.
- **hostname** – The name of the host to remove the groups from.
- **groups** – A string of a single group or a list or tuple of group names to remove from a host.

Returns True on complete success, False otherwise.

Return type bool

Raises `InvalidHost` – The host provided as argument does not exist.

external_users
Retrieves only users created by an external system.

Returns Users created by external system in tower.

Return type users (Generator)

get_all_groups_by_host_id(*host_id*)
Get groups for a particular host, which are directly and indirectly connected.

Parameters **host_id** – the id of the given host..

Returns list of custom groups.

Return type list

get_credential_by_id(*id_*)
Retrieves a credential by id.

Parameters **id** – The id of the credential to retrieve.

Returns The credential if a match is found else None.

Return type *Host*

get_credential_type_by_id(*id_*)
Retrieves a credential_type by id.

Parameters **id** – The id of the credential_type to retrieve.

Returns The credential_type if a match is found else None.

Return type *Host*

get_credential_type_by_name(*name*)
Retrieves a credential_type by name.

Parameters **name** – The name of the credential_type to retrieve.

Returns The credential_type if a match is found else None.

Return type *Host*

get_credentials_by_name(*name*)
Retrieves all credentials matching a certain name.

Parameters **name** – The name of the credential(s) to retrieve.

Returns A credentials generator.

Return type Credentials (Generator)

get_group_by_id(*id_*)
Retrieves a group by id.

Parameters **id** – The id of the group to retrieve.

Returns The group if a match is found else None.

Return type *Group*

get_host_by_id(*id_*)
Retrieves a host by id.

Parameters **id** – The id of the host to retrieve.

Returns The host if a match is found else None.

Return type *Host*

get_hosts_by_name(*name*)
Retrieves hosts by name.

Parameters **name** – The name of the hosts to retrieve.

Returns A generator with the matching hosts

Return type hosts (Generator)

get_inventories_by_name(*name*)
Retrieves inventories by name.

Parameters **name** – The name of the inventories to retrieve.

Returns A generator with the matching inventories

Return type inventories (Generator)

get_inventory_by_id(*id_*)
Retrieves an inventory by id.

Parameters **id** – The id of the inventory to retrieve.

Returns The inventory if a match is found else None.

Return type *Inventory*

get_inventory_group_by_name(*organization*, *inventory*, *name*)
Retrieves a group by name.

Parameters

- **organization** – The name of the organization the inventory belongs to.
- **inventory** – The inventory to retrieve the group from.
- **name** – The name of the group to retrieve.

Returns The group if a match is found else None.

Return type *Group*

Raises

- `InvalidOrganization` – The organisation provided as an argument does not exist.
- `InvalidInventory` – The inventory name provided as an argument does not exist.

get_inventory_host_by_name(*organization*, *inventory*, *name*)
Retrieves a host by name from an inventory.

Parameters

- **organization** – The name of the organization the inventory belongs to.
- **inventory** – The name of the inventory to search for a host.
- **name** – The name of the host to retrieve.

Returns The host if a match is found else None.

Return type *Host*

get_job_by_id(*id_*)
Retrieves a job by id.

Parameters **id** – The id of the job to retrieve.

Returns The host if a match is found else None.

Return type *Host*

get_job_template_by_id(*id_*)
Retrieves a job template by id.

Parameters **id** – The id of the job template to retrieve.

Returns The job template if a match is found else None.

Return type *Host*

get_job_template_by_name(*name*)
Retrieves job_template by name.

Parameters **name** – The name of the job_template to retrieve.

Returns A template with the matching name

Return type job_templates (*JobTemplate*)

get_jobs_by_name(*name*)
Get filtered list of jobs for a given name.

Parameters **name** – the given job name.

Returns the filtered list of jobs.

Return type list

get_organization_by_id(*id_*)
Retrieves an organization by id.

Parameters **id** – The id of the organization to retrieve.

Returns The organization if a match is found else None.

Return type *Organization*

get_organization_by_name(*name*)
Retrieves an organization by name.

Parameters **name** – The name of the organization to retrieve.

Returns The organization if a match is found else None.

Return type *Organization*

get_organization_credential_by_name(*organization*, *name*, *credential_type*)
Retrieves all credentials matching a certain name.

Parameters

- **organization** – The organization that owns the credential.
- **name** – The name of the credential(s) to retrieve.
- **credential_type** – The type of the credential.

Returns A credential if found else None.

Return type *Credential*

Raises

- `InvalidCredentialType` – The CredentialType given was not found.
- `InvalidOrganization` – The Organization given was not found.

get_organization_credential_by_name_with_type_id(*organization*, *name*, *credential_type_id*)
Retrieves all credentials matching a certain name.

Parameters

- **organization** (*str*) – The organization that owns the credential.
- **name** (*str*) – The name of the credential(s) to retrieve.
- **credential_type_id** (*int*) – The integer of the type of the credential.

Returns A credential if found else None.

Return type *Credential*

Raises `InvalidOrganization` – The Organization given was not found.

get_organization_inventory_by_name(*organization*, *name*)
Retrieves an inventory by name from an organization.

Parameters

- **organization** – The name of the organization to retrieve the inventory from.
- **name** – The name of the inventory to retrieve.

Returns The inventory if a match is found else None.

Return type *Inventory*

Raises `InvalidOrganization` – The organization provided as argument does not exist.

get_organization_inventory_script_by_name(*organization*, *name*)
Retrieves an custom inventory script by name from an organization.

Parameters

- **organization** – The name of the organization to retrieve the custom inventory script from.
- **name** – The name of the custom inventory script to retrieve.

Returns The custom inventory script if a match is found else None.

Return type *Inventory*

Raises `InvalidOrganization` – The organization provided as argument does not exist.

get_organization_project_by_name(*organization*, *name*)
Retrieves a project by name.

Parameters

- **organization** – The name of the organization the project belongs to.
- **name** – The name of the project to retrieve.

Returns The project if a match is found else None.

Return type *Project*

get_organization_team_by_name(*organization*, *name*)
Retrieves a team by name.

Parameters

- **organization** – The name of the organization the team belongs to.
- **name** – The name of the team to retrieve.

Returns The team if a match is found else None.

Return type *Team*

Raises `InvalidOrganization` – The organization provided as argument does not exist.

get_project_by_id(*id_*)
Retrieves a project by id.

Parameters **id** – The id of the project to retrieve.

Returns The project if a match is found else None.

Return type *Project*

get_project_update_by_id(*id_*)
Retrieves a project_update by id.

Parameters **id** – The id of the project_update to retrieve.

Returns The project_update if a match is found else None.

Return type *Host*

get_project_updates_by_name(*name*)
Retrieves project_updates matching a certain name.

Parameters **name** – the given job_update name.

Returns the filtered list of project update jobs.

Return type list

get_projects_by_name(*name*)
Retrieves projects by name.

Parameters **name** – The name of the projects to retrieve.

Returns A generator with the matching projects

Return type projects (Generator)

get_schedule_by_id(*id_*)
Retrieves a schedule by id.

Parameters **id** – The id of the schedule to retrieve.

Returns The schedule if a match is found else None.

Return type *Schedule*

get_schedule_by_name(*name*)
Retrieves an schedule by name.

Parameters **name** – The name of the schedule to retrieve.

Returns The schedule if a match is found else None.

Return type *Schedule*

get_system_job_by_id(*id_*)
Retrieves a job by id.

Parameters **id** – The id of the job to retrieve.

Returns The job if a match is found else None.

Return type *Host*

get_system_jobs_by_name(*name*)
Retrieves all system jobs matching a certain name.

Parameters **name** – The name of the system job(s) to retrieve.

Returns A system job generator.

Return type UnifiedJob (Generator)

get_team_by_id(*id_*)
Retrieves a team by id.

Parameters **id** – The id of the team to retrieve.

Returns The team if a match is found else None.

Return type *Team*

get_teams_by_name(*name*)
Retrieves teams by name.

Parameters **name** – The name of the teams to retrieve.

Returns A generator with the matching teams

Return type teams (Generator)

get_unified_job_by_id(*id_*)
Retrieves a job by id.

Parameters **id** – The id of the job to retrieve.

Returns The job if a match is found else None.

Return type *Host*

get_unified_jobs_by_name(*name*)
Retrieves all unified jobs matching a certain name.

Parameters **name** – The name of the unified job(s) to retrieve.

Returns A unified job generator.

Return type UnifiedJob (Generator)

get_user_by_id(*id_*)
Retrieves a user by id.

Parameters **id** – The id of the user to retrieve.

Returns The user if a match is found else None.

Return type *User*

get_user_by_username(*name*)
Retrieves user by name.

Parameters **name** – The name of the user to retrieve.

Returns The user if a match is found else None.

Return type user (*User*)

get_workflow_job_by_id(*id_*)
Retrieves a job by id.

Parameters **id** – The id of the job to retrieve.

Returns The job if a match is found else None.

Return type *Host*

get_workflow_job_template_by_id(*id_*)
Retrieves a workflow template job by id.

Parameters **id** – The id of the workflow template job to retrieve.

Returns The job if a match is found else None.

Return type *Host*

get_workflow_job_templates_by_name(*name*)
Retrieves all workflow template jobs matching a certain name.

Parameters **name** – The name of the workflow template job(s) to retrieve.

Returns A workflow template job generator.

Return type UnifiedJob (Generator)

get_workflow_jobs_by_name(*name*)
Retrieves all workflow jobs matching a certain name.

Parameters **name** – The name of the workflow job(s) to retrieve.

Returns A workflow job generator.

Return type UnifiedJob (Generator)

groups
The groups configured in tower.

Returns The manager object for groups.

Return type *EntityManager*

hosts
The hosts configured in tower.

Returns The manager object for hosts

Return type *EntityManager*

instance_groups
The instance_groups configured in tower.

Returns The manager object for instance groups.

Return type *EntityManager*

instances
The instances configured in tower.

Returns The manager object for instances.

Return type *EntityManager*

inventories
The inventories configured in tower.

Returns The inventories configured in tower.‘

Return type list of Inventory

inventory_scripts
The inventories configured in tower.

Returns The inventories configured in tower.‘

Return type list of Inventory

inventory_sources
A manager object for the inventory_sources in tower.

Returns The manager object for inventory_sources.

Return type *EntityManager*

job_events
The job templates configured in tower.

Returns The manager object for job templates.

Return type *EntityManager*

job_templates
The job templates configured in tower.

Returns The manager object for job templates.

Return type *EntityManager*

jobs
The jobs executed in tower.

Returns The manager object for jobs.

Return type *EntityManager*

local_users
Retrieves only users created locally in tower.

Returns Users created locally in tower.

Return type users (Generator)

notification_templates
The notification templates configured in tower.

Returns The manager object for groups.

Return type *EntityManager*

organizations
The organizations configured in tower.

Returns The manager object for organizations.

Return type *EntityManager*

project_updates
A manager object for the project_updates in tower.

Returns A generator of project updates.

Return type project_updates (*EntityManager*)

projects
The projects configured in tower.

Returns The manager object for projects.

Return type *EntityManager*

roles
The roles configured in tower.

Returns The manager object for roles.

Return type *EntityManager*

schedules
The schedules configured in tower.

Returns The manager object for schedules.

Return type *EntityManager*

settings
The settings part of tower.

Returns The manager object for settings.

Return type *EntityManager*

system_jobs
The system jobs executed in tower.

Returns The manager object for system jobs.

Return type *EntityManager*

teams
The teams configured in tower.

Returns The manager object for teams.

Return type *EntityManager*

tower_credential_types
The default credential_types configured in tower.

Returns The manager object for internal credential types.

Return type *EntityManager*

unified_job_templates
The unified job templates configured in tower.

Returns The manager object for unified job templates.

Return type *EntityManager*

unified_jobs
The unified jobs executed in tower.

Returns The manager object for unified jobs.

Return type *EntityManager*

update_all_organization_projects(*organization_name*)
Update all the projects in ansible tower for a given organization.

Parameters **organization_name** – The name of the organization.

update_organization_project_by_name(*organization_name*, *project_name*)
Update the ansible tower project with given project name.

Parameters

- **organization_name** – The name of the organization.
- **project_name** – The name of the project, which is to be updated.

Returns dict of response of api request as json on success, None otherwise.

Return type API Response (dict)

update_organization_projects_by_branch_name(*scm_url*, *branch_name*, *organization_name*)
Update an ansible tower project or list of projects for an organization based on their branch name.

A scm_branch can only be identified correctly with a corresponding scm_url.

Parameters

- **scm_url** – the URL of the relevant repository configured in the project.
- **branch_name** – the name of the branch, which is selected as scm_branch parameter of the project.
- **organization_name** – the name of the organization.

update_organization_projects_by_scm_url (*scm_url*, *organization_name*)
Send update request to update project for a given git repository (scm_url) withing an organization.

Parameters

- **organization_name** – the name of the organization.
- **scm_url** – the http url of the required repository.

update_project_by_id (*project_id*)
Update the ansible tower project with given project id.

Parameters **project_id** – The id of the project, which is to be updated.

Returns List of response of api request as json on success, None otherwise.

Return type list

users
A manager object for the users in tower.

Returns The manager object for users.

Return type *EntityManager*

workflow_job_templates
The workflow job templates configured in tower.

Returns The manager object for workflow job templates.

Return type *EntityManager*

workflow_jobs
The workflow jobs executed in tower.

Returns The manager object for workflow jobs.

Return type *EntityManager*

5.1.4 towerlib.towerlibexceptions module

Custom exception code for towerlib.

exception towerlib.towerlibexceptions.**AuthFailed**
Bases: Exception

The token retrieval failed.

exception towerlib.towerlibexceptions.**FailedToDeleteTemplate**
Bases: Exception

The deletion of the job template failed.

exception towerlib.towerlibexceptions.**InvalidCredential**
Bases: Exception

The credential provided is invalid.

exception towerlib.towerlibexceptions.**InvalidCredentialType**
Bases: Exception

The credential type provided is invalid.

exception towerlib.towerlibexceptions.**InvalidGroup**
Bases: Exception

The group provided is invalid.

exception towerlib.towerlibexceptions.**InvalidHost**
Bases: Exception

The host provided is invalid.

exception towerlib.towerlibexceptions.**InvalidInstanceGroup**
Bases: Exception

The instance group provided does not exist.

exception towerlib.towerlibexceptions.**InvalidInventory**
Bases: Exception

The inventory provided is invalid.

exception towerlib.towerlibexceptions.**InvalidInventoryScript**
Bases: Exception

The inventory script provided is invalid.

exception towerlib.towerlibexceptions.**InvalidJobTemplate**
Bases: Exception

The job template provided is not valid.

exception towerlib.towerlibexceptions.**InvalidJobType**
Bases: Exception

The job type provided is not valid. Valid values (u'run', u'check').

exception towerlib.towerlibexceptions.**InvalidOrganization**
Bases: Exception

The organization provided is not a valid organization.

exception towerlib.towerlibexceptions.**InvalidPlaybook**
Bases: Exception

The playbook specified does not exist in the project.

exception towerlib.towerlibexceptions.**InvalidProject**
Bases: Exception

The project provided is not valid.

exception towerlib.towerlibexceptions.**InvalidRole**
Bases: Exception

The role is not valid for this organization.

exception towerlib.towerlibexceptions.**InvalidSchedule**
Bases: Exception

The schedule is not valid for this job template.

exception towerlib.towerlibexceptions.**InvalidTeam**
Bases: Exception

The team provided is invalid.

exception towerlib.towerlibexceptions.**InvalidUser**
Bases: Exception

The user provided is invalid.

exception towerlib.towerlibexceptions.**InvalidUserLevel**
Bases: Exception

The value provided is not allowed.

Valid values ('standard', 'system_auditor', 'system_administrator')

exception towerlib.towerlibexceptions.**InvalidValue**
Bases: Exception

The value is not valid for the field.

exception towerlib.towerlibexceptions.**InvalidVariables**
Bases: Exception

The variables are not valid json.

exception towerlib.towerlibexceptions.**InvalidVerbosity**
Bases: Exception

The verbosity level provided is not valid. Valid values (0, 1, 2, 3, 4).

exception towerlib.towerlibexceptions.**PermissionNotFound**
Bases: Exception

The permission was not found in the entity.

5.1.5 Module contents

towerlib package.

Import all parts from towerlib here.

CHAPTER 6

Credits

6.1 Development Lead

- Costas Tyfoxylos <ctyfoxylos@schubergphilis.com>

6.2 Contributors

- Michael Wiley <michaeltwiley@gmail.com>
- Anton Opgenoort <aopgenoort@schubergphilis.com>
- Gareth Hawker <ghawker@schubergphilis.com>
- Ilija Matoski <imatoski@schubergphilis.com>
- Yorick Hoorneman <yhoorneman@schubergphilis.com>
- Howard Jones <howie@thingy.com>
- Octavian Balas <octavian.balas@gmail.com>
- Philipp Hoffmann <phoffmann@spirit21.com>
- Iqbal Nazir <iqbalnazirsumon@gmail.com>
- Jeremy Shade <jeremy.shade@outlook.com>
- Yuki Yamashita <kono@ms1.kononet.jp>

CHAPTER 7

History

CHAPTER 8

0.1.0 (25-05-2018)

- First release

CHAPTER 9

0.2.0 (27-07-2018)

- Refactored code to use entity managers for all tower objects saving a huge amount of network calls and implemented filtering
- Removed pipenv locking mechanism as this is broken for python 2.7 completely

CHAPTER 10

0.3.0 (01-08-2018)

- Added capability to launch job template job

CHAPTER 11

1.0.0 (27-09-2018)

- Added support for specifying http or https and certificate verifications options
- Extented the editing capabilities of hosts to name, description and enabled status

CHAPTER 12

2.0.0 (16-10-2018)

- Implemented dynamic attributes in running jobs.
- Implemented cancel capabilities for running jobs.
- Updated the template to python 3.7
- Officially dropped support for python 2.7

CHAPTER 13

2.0.1 (25-10-2018)

- Update template and dependencies

CHAPTER 14

2.0.2 (25-10-2018)

- Reverted breaking change for upload script

CHAPTER 15

2.0.3 (29-11-2018)

- Fixed reference in the package for the right github repo

CHAPTER 16

2.1.0 (29-11-2018)

- Fixed issue https://github.com/schubergphilis/towerlib/issues/11 with some organization entities

CHAPTER 17

2.2.0 (03-12-2018)

- Implemented group association retrieval (contribution by <aopgenoort@schubergphilis.com>)

CHAPTER 18

2.3.0 (05-12-2018)

- Added group association and disassociation

CHAPTER 19

2.3.1 (03-01-2019)

- Bumped Requests

CHAPTER 20

2.3.2 (09-01-2019)

- Changed library.py

CHAPTER 21

2.3.3 (07-03-2019)

- update setup.py

CHAPTER 22

2.3.4 (04-04-2019)

- Added missing import of object that caused a bug in the EntityManager crashing, not being able to load it.

CHAPTER 23

2.3.5 (25-07-2019)

- fix for https://github.com/schubergphilis/towerlib/pull/22

CHAPTER 24

2.3.6 (31-07-2019)

- Fixed inventory host deletion.

CHAPTER 25

3.0.0 (18-10-2019)

- Fixed the references to all the objects to be identified by their parent relationship to avoid ambiguity. Full test coverage.

CHAPTER 26

3.0.1 (18-10-2019)

- bumped dependencies

CHAPTER 27

3.1.0 (18-10-2019)

- Implemented basic inventory source addition functionality

CHAPTER 28

3.2.0 (18-10-2019)

- Implemented helper method to create inventory source with an existing credential id.

CHAPTER 29

3.2.1 (01-11-2019)

- fixed add_credential

CHAPTER 30

3.2.2 (08-11-2019)

- Added case incensitive search

CHAPTER 31

3.2.3 (06-12-2019)

- Fix pagination for filtering.

CHAPTER 32

3.2.4 (12-12-2019)

- fixed pagination!!

CHAPTER 33

3.2.5 (19-12-2019)

- Fixed underlying _update_values method to prevent overwrite of nested dicts in AWX 9.0.1.

CHAPTER 34

3.2.6 (19-12-2019)

- Linting and pipfile dependency fixes.

CHAPTER 35

3.2.7 (14-03-2020)

- Added support for python 3.6 by adding dataclasses package with marker

CHAPTER 36

3.2.8 (14-03-2020)

- Removed unwanted development files from the final package.

CHAPTER 37

3.2.9 (29-05-2020)

- Added support for Schedules

CHAPTER 38

3.2.10 (09-06-2020)

- bumped requests

CHAPTER 39

3.3.0 (06-07-2020)

- Implemented capabilities to add organization object roles to team permissions.

CHAPTER 40

3.4.0 (09-10-2020)

- bump requests

CHAPTER 41

3.4.1 (16-10-2020)

- Bumped dependencies

CHAPTER 42

3.4.2 (06-11-2020)

- Added missing development dependency and bumped dependencies to latest patch versions.

CHAPTER 43

3.5.0 (02-12-2020)

- Changed the credential creation process to make team and Owner name optional

CHAPTER 44

3.6.0 (06-01-2021)

- Extended JobTemplate with some methods and made credential not required on job template creation.

CHAPTER 45

3.6.1 (06-01-2021)

- Loosened up dependencies.

CHAPTER 46

3.6.2 (04-02-2021)

- Fix missing f-string.

CHAPTER 47

3.7.0 (09-03-2021)

- Implemented survey specification retrieval.

CHAPTER 48

3.7.1 (26-04-2021)

- Bumped dependencies.

CHAPTER 49

3.7.2 (26-04-2021)

- Bumped dependencies.

CHAPTER 50

3.7.3 (02-06-2021)

- Bumped dependencies and fixed a couple of bugs on job attribute retrieval.

CHAPTER 51

3.8.0 (01-07-2021)

- Features implemented by spirit21.

CHAPTER 52

3.9.0 (01-07-2021)

- Features implemented by spirit21.

CHAPTER 53

3.9.1 (10-09-2021)

- Added the capability to launch a job template with non-default SCM branches.

CHAPTER 54

3.9.2 (10-09-2021)

- Linted.

CHAPTER 55

3.9.3 (11-07-2022)

- Fixed bug with job execution.

CHAPTER 56

3.9.4 (01-09-2022)

- Fixed some redirects, bumped dependencies.

CHAPTER 57

3.10.0 (04-10-2022)

- Allows unsetting of default values in launch job courtesy of https://github.com/EliasBoulharts

CHAPTER 58

3.11.0 (11-11-2022)

- added summary_fields in JobTemplate

CHAPTER 59

3.12.0 (10-03-2023)

- Implement aws credentials and expose input sources.

CHAPTER 60

3.13.0 (25-04-2023)

- Add default timeout.

CHAPTER 61

3.13.1 (05-05-2023)

- Fix for unbound local error.

CHAPTER 62

3.14.0 (13-09-2023)

- Implement workflow nodes curtesy of Yuki Yamashita <kono@ms1.kononet.jp>.

CHAPTER 63

Indices and tables

- genindex
- modindex
- search

Python Module Index

t

towerlib, 101
towerlib.entities, 79
towerlib.entities.core, 13
towerlib.entities.credential, 17
towerlib.entities.group, 21
towerlib.entities.host, 23
towerlib.entities.instance, 25
towerlib.entities.inventory, 27
towerlib.entities.inventory_script, 32
towerlib.entities.inventory_source, 33
towerlib.entities.job, 34
towerlib.entities.notification, 53
towerlib.entities.organization, 59
towerlib.entities.project, 64
towerlib.entities.role, 67
towerlib.entities.schedule, 68
towerlib.entities.settings, 70
towerlib.entities.team, 72
towerlib.entities.user, 77
towerlib.towerlib, 79
towerlib.towerlibexceptions, 99

Index

A

access_key (*towerlib.entities.credential.AmazonWebServicesCredential attribute*), 18
account_sid (*towerlib.entities.notification.NotificationTwilio attribute*), 58
account_token (*towerlib.entities.notification.NotificationTwilio attribute*), 58
activity_streams (*towerlib.entities.core.LicenseFeatures attribute*), 16
add_credential() (*towerlib.entities.job.JobTemplate method*), 41
add_credential_by_id() (*towerlib.entities.job.JobTemplate method*), 42
add_credential_permission_admin() (*towerlib.entities.team.Team method*), 72
add_credential_permission_use() (*towerlib.entities.team.Team method*), 73
add_host_by_name() (*towerlib.entities.group.Group method*), 21
add_inventory_permission_ad_hoc() (*towerlib.entities.team.Team method*), 73
add_inventory_permission_admin() (*towerlib.entities.team.Team method*), 73
add_inventory_permission_update() (*towerlib.entities.team.Team method*), 73
add_inventory_permission_use() (*towerlib.entities.team.Team method*), 73
add_job_template_permission_admin() (*towerlib.entities.team.Team method*), 73
add_job_template_permission_execute() (*towerlib.entities.team.Team method*), 73
add_organization_role_by_name() (*towerlib.entities.team.Team method*), 73
add_project_permission_admin() (*towerlib.entities.team.Team method*), 74
add_project_permission_update() (*towerlib.entities.team.Team method*), 74
add_project_permission_use() (*towerlib.entities.team.Team method*), 74
add_schedule() (*towerlib.entities.job.JobTemplate method*), 42
add_slash() (*towerlib.towerlib.Tower static method*), 79
add_user_as_admin() (*towerlib.entities.team.Team method*), 74
add_user_as_member() (*towerlib.entities.team.Team method*), 74
AdHocCommandJob (*class in towerlib.entities.job*), 34
admins_count (*towerlib.entities.organization.Organization attribute*), 59
all_groups (*towerlib.entities.host.Host attribute*), 23
allow_simultaneous (*towerlib.entities.job.JobTemplate attribute*), 42
AmazonWebServicesCredential (*class in towerlib.entities.credential*), 17
analytics_status (*towerlib.entities.core.Config attribute*), 14
ansible_facts (*towerlib.entities.host.Host attribute*), 23
ansible_version (*towerlib.entities.core.Config attribute*), 14
api_url (*towerlib.entities.core.Entity attribute*), 15
api_url (*towerlib.entities.notification.NotificationHipChat attribute*), 55
ask_credential_on_launch (*towerlib.entities.job.JobTemplate attribute*), 42
ask_diff_mode_on_launch (*towerlib.entities.job.JobTemplate attribute*), 42
ask_inventory_on_launch (*towerlib.entities.job.JobTemplate attribute*), 42
ask_job_type_on_launch (*towerlib.entities.job.JobTemplate attribute*), 42
ask_limit_on_launch (*towerlib.entities.job.JobTemplate attribute*), 42
ask_skip_tags_on_launch (*tower-*

lib.entities.job.JobTemplate attribute), 43
ask_tags_on_launch (*towerlib.entities.job.JobTemplate attribute*), 43
ask_variables_on_launch (*towerlib.entities.job.JobTemplate attribute*), 43
ask_verbosity_on_launch (*towerlib.entities.job.JobTemplate attribute*), 43
associate_group_by_name() (*towerlib.entities.group.Group method*), 21
associate_groups_with_inventory_host() (*towerlib.towerlib.Tower method*), 79
associate_with_groups() (*towerlib.entities.host.Host method*), 23
associate_with_organization_role() (*towerlib.entities.user.User method*), 77
auth (*towerlib.entities.user.User attribute*), 77
AuthFailed, 99
available_instances (*towerlib.entities.core.LicenseInfo attribute*), 16

B

become_enabled (*towerlib.entities.job.AdHocCommandJob attribute*), 35
become_enabled (*towerlib.entities.job.JobTemplate attribute*), 43

C

ca_host_verify (*towerlib.entities.credential.HashicorpVaultCredential attribute*), 20
callback_url (*towerlib.entities.settings.Saml attribute*), 70
cancel() (*towerlib.entities.job.JobRun method*), 38
cancel() (*towerlib.entities.job.WorkflowJobRun method*), 52
capacity (*towerlib.entities.core.ClusterInstance attribute*), 13
capacity (*towerlib.entities.instance.Instance attribute*), 25
capacity (*towerlib.entities.instance.InstanceGroup attribute*), 26
channels (*towerlib.entities.notification.NotificationSlack attribute*), 57
client_name (*towerlib.entities.notification.NotificationPagerDuty attribute*), 56
Cluster (*in module towerlib.entities.core*), 13
cluster (*towerlib.towerlib.Tower attribute*), 80
ClusterInstance (*class in towerlib.entities.core*), 13
color (*towerlib.entities.notification.NotificationHipChat attribute*), 55
company_name (*towerlib.entities.core.LicenseInfo attribute*), 16
compliant (*towerlib.entities.core.LicenseInfo attribute*), 16
Config (*class in towerlib.entities.core*), 14
configuration (*towerlib.towerlib.Tower attribute*), 80
consumed_capacity (*towerlib.entities.core.ClusterInstance attribute*), 13
consumed_capacity (*towerlib.entities.instance.Instance attribute*), 25
consumed_capacity (*towerlib.entities.instance.InstanceGroup attribute*), 26
controller (*towerlib.entities.instance.InstanceGroup attribute*), 26
count (*towerlib.entities.job.WorkflowNodes attribute*), 52
counter (*towerlib.entities.job.JobEvent attribute*), 35
create_credential_in_organization() (*towerlib.towerlib.Tower method*), 80
create_credential_in_organization_with_type_id() (*towerlib.towerlib.Tower method*), 80
create_credential_type() (*towerlib.towerlib.Tower method*), 81
create_credential_with_credential_type_id() (*towerlib.towerlib.Tower method*), 81
create_group() (*towerlib.entities.inventory.Inventory method*), 27
create_host() (*towerlib.entities.inventory.Inventory method*), 27
create_host_in_inventory() (*towerlib.towerlib.Tower method*), 82
create_inventory() (*towerlib.entities.organization.Organization method*), 59
create_inventory_group() (*towerlib.towerlib.Tower method*), 82
create_inventory_script() (*towerlib.entities.organization.Organization method*), 59
create_job_template() (*towerlib.towerlib.Tower method*), 82
create_organization() (*towerlib.towerlib.Tower method*), 84
create_organization_inventory() (*towerlib.towerlib.Tower method*), 84
create_organization_inventory_script() (*towerlib.towerlib.Tower method*), 84
create_project() (*towerlib.entities.organization.Organization method*), 60
create_project_in_organization() (*towerlib.towerlib.Tower method*), 84

create_source() (towerlib.entities.inventory.Inventory method), 28
create_source_with_credential_id() (towerlib.entities.inventory.Inventory method), 28
create_team() (towerlib.entities.organization.Organization method), 60
create_team_in_organization() (towerlib.towerlib.Tower method), 85
create_user() (towerlib.towerlib.Tower method), 85
create_user_in_organization() (towerlib.towerlib.Tower method), 86
created_at (towerlib.entities.core.ClusterInstance attribute), 13
created_at (towerlib.entities.core.Entity attribute), 15
created_at (towerlib.entities.job.JobEvent attribute), 36
created_by (towerlib.entities.credential.GenericCredential attribute), 19
created_by (towerlib.entities.group.Group attribute), 21
created_by (towerlib.entities.host.Host attribute), 23
created_by (towerlib.entities.inventory.Inventory attribute), 29
created_by (towerlib.entities.job.JobRun attribute), 38
created_by (towerlib.entities.job.JobTemplate attribute), 43
created_by (towerlib.entities.organization.Organization attribute), 60
created_by (towerlib.entities.project.Project attribute), 64
Credential (class in towerlib.entities.credential), 18
credential (towerlib.entities.job.AdHocCommandJob attribute), 35
credential (towerlib.entities.project.Project attribute), 64
credential_type (towerlib.entities.credential.GenericCredential attribute), 19
credential_types (towerlib.towerlib.Tower attribute), 86
credentials (towerlib.entities.job.JobRun attribute), 38
credentials (towerlib.entities.job.JobTemplate attribute), 43
credentials (towerlib.entities.job.ProjectUpdateJob attribute), 47
credentials (towerlib.entities.organization.Organization attribute), 60
credentials (towerlib.entities.team.Team attribute), 74
credentials (towerlib.entities.user.User attribute), 77
credentials (towerlib.towerlib.Tower attribute), 86
CredentialType (class in towerlib.entities.credential), 18
current_instances (towerlib.entities.core.LicenseInfo attribute), 16
custom_credential_types (towerlib.towerlib.Tower attribute), 86
custom_virtualenv (towerlib.entities.organization.Organization attribute), 61
custom_virtualenv (towerlib.entities.project.Project attribute), 64

D

date_expired (towerlib.entities.core.LicenseInfo attribute), 16
date_warning (towerlib.entities.core.LicenseInfo attribute), 16
DateParserMixin (class in towerlib.entities.core), 14
datetime_end (towerlib.entities.schedule.Schedule attribute), 68
datetime_start (towerlib.entities.schedule.Schedule attribute), 68
DEFAULT_MEMBER_ROLE (towerlib.entities.organization.Organization attribute), 59
delete() (towerlib.entities.core.Entity method), 15
delete_credential_type() (towerlib.towerlib.Tower method), 86
delete_group() (towerlib.entities.inventory.Inventory method), 29
delete_host() (towerlib.entities.inventory.Inventory method), 29
delete_inventory() (towerlib.entities.organization.Organization method), 61
delete_inventory_group() (towerlib.towerlib.Tower method), 87
delete_inventory_host() (towerlib.towerlib.Tower method), 87
delete_inventory_script() (towerlib.entities.organization.Organization method), 61
delete_job_template() (towerlib.towerlib.Tower method), 87
delete_organization() (towerlib.towerlib.Tower method), 87
delete_organization_credential_by_name() (towerlib.towerlib.Tower method), 87
delete_organization_credential_by_name_with_type_id() (towerlib.towerlib.Tower method), 88

delete_organization_inventory() (*towerlib.towerlib.Tower method*), 88
delete_organization_inventory_script() (*towerlib.towerlib.Tower method*), 88
delete_organization_project() (*towerlib.towerlib.Tower method*), 88
delete_project() (*towerlib.entities.organization.Organization method*), 61
delete_team() (*towerlib.entities.organization.Organization method*), 61
delete_team_in_organization() (*towerlib.towerlib.Tower method*), 89
delete_user() (*towerlib.towerlib.Tower method*), 89
deployment_id (*towerlib.entities.core.LicenseInfo attribute*), 17
description (*towerlib.entities.credential.CredentialType attribute*), 18
description (*towerlib.entities.credential.GenericCredential attribute*), 19
description (*towerlib.entities.group.Group attribute*), 21
description (*towerlib.entities.host.Host attribute*), 23
description (*towerlib.entities.inventory.Inventory attribute*), 30
description (*towerlib.entities.inventory_script.InventoryScript attribute*), 32
description (*towerlib.entities.inventory_source.InventorySource attribute*), 33
description (*towerlib.entities.job.JobRun attribute*), 38
description (*towerlib.entities.job.JobTemplate attribute*), 43
description (*towerlib.entities.job.ProjectUpdateJob attribute*), 47
description (*towerlib.entities.job.SystemJob attribute*), 50
description (*towerlib.entities.notification.NotificationTemplate attribute*), 57
description (*towerlib.entities.organization.Organization attribute*), 61
description (*towerlib.entities.project.Project attribute*), 64
description (*towerlib.entities.role.Role attribute*), 67
description (*towerlib.entities.schedule.Schedule attribute*), 68
description (*towerlib.entities.team.Team attribute*), 74
diff_mode (*towerlib.entities.job.AdHocCommandJob attribute*), 35
diff_mode (*towerlib.entities.job.JobTemplate attribute*), 43
diff_mode (*towerlib.entities.schedule.Schedule attribute*), 69
disable_ssl_verification (*towerlib.entities.notification.NotificationWebHook attribute*), 59
disassociate_from_organization_role() (*towerlib.entities.user.User method*), 77
disassociate_group_by_name() (*towerlib.entities.group.Group method*), 21
disassociate_groups_from_inventory_host() (*towerlib.towerlib.Tower method*), 89
disassociate_with_groups() (*towerlib.entities.host.Host method*), 24

E

elapsed (*towerlib.entities.job.ProjectUpdateJob attribute*), 47
elapsed (*towerlib.entities.job.SystemJob attribute*), 50
elapsed_time (*towerlib.entities.job.JobRun attribute*), 38
email (*towerlib.entities.user.User attribute*), 77
enabled (*towerlib.entities.host.Host attribute*), 24
enabled (*towerlib.entities.schedule.Schedule attribute*), 69
enabled_idps (*towerlib.entities.settings.Saml attribute*), 70
end_line (*towerlib.entities.job.JobEvent attribute*), 36
enterprise_auth (*towerlib.entities.core.LicenseFeatures attribute*), 16
Entity (*class in towerlib.entities.core*), 14
EntityManager (*class in towerlib.entities.core*), 15
error (*towerlib.entities.notification.Notification attribute*), 53
eula (*towerlib.entities.core.Config attribute*), 14
event (*towerlib.entities.job.JobEvent attribute*), 36
event_data (*towerlib.entities.job.JobEvent attribute*), 36
event_display (*towerlib.entities.job.JobEvent attribute*), 36
event_level (*towerlib.entities.job.JobEvent attribute*), 36
execution_node (*towerlib.entities.job.ProjectUpdateJob attribute*), 47
execution_node (*towerlib.entities.job.SystemJob attribute*), 50

external_account (*towerlib.entities.user.User attribute*), 78
external_users (*towerlib.towerlib.Tower attribute*), 89
extra_credentials (*towerlib.entities.job.JobRun attribute*), 38
extra_credentials (*towerlib.entities.job.JobTemplate attribute*), 43
extra_data (*towerlib.entities.schedule.Schedule attribute*), 69
extra_data (*towerlib.entities.settings.Saml attribute*), 71
extra_vars (*towerlib.entities.job.JobRun attribute*), 38
extra_vars (*towerlib.entities.job.JobTemplate attribute*), 44
extra_vars (*towerlib.entities.job.SystemJob attribute*), 50

F

failed (*towerlib.entities.job.JobSummary attribute*), 40
FailedToDeleteTemplate, 99
failures_count (*towerlib.entities.job.JobSummary attribute*), 40
features (*towerlib.entities.core.LicenseInfo attribute*), 17
filter() (*towerlib.entities.core.EntityManager method*), 15
finished (*towerlib.entities.job.JobRun attribute*), 39
finished_at (*towerlib.entities.job.ProjectUpdateJob attribute*), 47
finished_at (*towerlib.entities.job.SystemJob attribute*), 50
first_name (*towerlib.entities.user.User attribute*), 78
force_handlers (*towerlib.entities.job.JobTemplate attribute*), 44
forks_count (*towerlib.entities.job.AdHocCommandJob attribute*), 35
forks_count (*towerlib.entities.job.JobTemplate attribute*), 44
free_instances (*towerlib.entities.core.LicenseInfo attribute*), 17
from_number (*towerlib.entities.notification.NotificationTwilio attribute*), 58

G

GenericCredential (*class in towerlib.entities.credential*), 19
get_all_groups_by_host_id() (*towerlib.towerlib.Tower method*), 89
get_credential_by_id() (*towerlib.entities.organization.Organization method*), 61
get_credential_by_id() (*towerlib.towerlib.Tower method*), 89
get_credential_by_name() (*towerlib.entities.organization.Organization method*), 61
get_credential_by_name_with_type_id() (*towerlib.entities.organization.Organization method*), 62
get_credential_type_by_id() (*towerlib.towerlib.Tower method*), 90
get_credential_type_by_name() (*towerlib.towerlib.Tower method*), 90
get_credentials_by_name() (*towerlib.towerlib.Tower method*), 90
get_group_by_id() (*towerlib.towerlib.Tower method*), 90
get_group_by_name() (*towerlib.entities.inventory.Inventory method*), 30
get_host_by_id() (*towerlib.towerlib.Tower method*), 90
get_host_by_name() (*towerlib.entities.inventory.Inventory method*), 30
get_hosts_by_name() (*towerlib.towerlib.Tower method*), 90
get_inventories_by_name() (*towerlib.towerlib.Tower method*), 90
get_inventory_by_id() (*towerlib.towerlib.Tower method*), 91
get_inventory_by_name() (*towerlib.entities.organization.Organization method*), 62
get_inventory_group_by_name() (*towerlib.towerlib.Tower method*), 91
get_inventory_host_by_name() (*towerlib.towerlib.Tower method*), 91
get_inventory_script_by_name() (*towerlib.entities.organization.Organization method*), 62
get_job_by_id() (*towerlib.towerlib.Tower method*), 91
get_job_template_by_id() (*towerlib.towerlib.Tower method*), 91
get_job_template_by_name() (*towerlib.towerlib.Tower method*), 91
get_jobs_by_name() (*towerlib.towerlib.Tower method*), 92
get_organization_by_id() (*towerlib.towerlib.Tower method*), 92
get_organization_by_name() (*towerlib.towerlib.Tower method*), 92
get_organization_credential_by_name()

(*towerlib.towerlib.Tower method*), 92
get_organization_credential_by_name_with_type_id() (*towerlib.towerlib.Tower method*), 92
get_organization_inventory_by_name() (*towerlib.towerlib.Tower method*), 92
get_organization_inventory_script_by_name() (*towerlib.towerlib.Tower method*), 93
get_organization_project_by_name() (*towerlib.towerlib.Tower method*), 93
get_organization_team_by_name() (*towerlib.towerlib.Tower method*), 93
get_project_by_id() (*towerlib.towerlib.Tower method*), 93
get_project_by_name() (*towerlib.entities.organization.Organization method*), 62
get_project_update_by_id() (*towerlib.towerlib.Tower method*), 93
get_project_updates_by_name() (*towerlib.towerlib.Tower method*), 94
get_projects_by_name() (*towerlib.towerlib.Tower method*), 94
get_schedule_by_id() (*towerlib.towerlib.Tower method*), 94
get_schedule_by_name() (*towerlib.towerlib.Tower method*), 94
get_system_job_by_id() (*towerlib.towerlib.Tower method*), 94
get_system_jobs_by_name() (*towerlib.towerlib.Tower method*), 94
get_team_by_id() (*towerlib.towerlib.Tower method*), 94
get_team_by_name() (*towerlib.entities.organization.Organization method*), 62
get_teams_by_name() (*towerlib.towerlib.Tower method*), 95
get_unified_job_by_id() (*towerlib.towerlib.Tower method*), 95
get_unified_jobs_by_name() (*towerlib.towerlib.Tower method*), 95
get_user_by_id() (*towerlib.towerlib.Tower method*), 95
get_user_by_username() (*towerlib.entities.team.Team method*), 74
get_user_by_username() (*towerlib.towerlib.Tower method*), 95
get_workflow_job_by_id() (*towerlib.towerlib.Tower method*), 95
get_workflow_job_template_by_id() (*towerlib.towerlib.Tower method*), 95
get_workflow_job_templates_by_name() (*towerlib.towerlib.Tower method*), 95
get_workflow_jobs_by_name() (*towerlib.towerlib.Tower method*), 96
grace_period_remaining (*towerlib.entities.core.LicenseInfo attribute*), 17
grafana_key (*towerlib.entities.notification.NotificationGrafana attribute*), 54
grafana_url (*towerlib.entities.notification.NotificationGrafana attribute*), 54
Group (*class in towerlib.entities.group*), 21
groups (*towerlib.entities.group.Group attribute*), 21
groups (*towerlib.entities.host.Host attribute*), 24
groups (*towerlib.entities.inventory.Inventory attribute*), 30
groups (*towerlib.towerlib.Tower attribute*), 96
groups_with_active_failures_count (*towerlib.entities.group.Group attribute*), 22
groups_with_active_failures_count (*towerlib.entities.inventory.Inventory attribute*), 30

H

ha (*towerlib.entities.core.LicenseFeatures attribute*), 16
has_active_failures (*towerlib.entities.group.Group attribute*), 22
has_active_failures (*towerlib.entities.host.Host attribute*), 24
has_active_failures (*towerlib.entities.inventory.Inventory attribute*), 30
has_inventory_sources (*towerlib.entities.group.Group attribute*), 22
has_inventory_sources (*towerlib.entities.host.Host attribute*), 24
has_inventory_sources (*towerlib.entities.inventory.Inventory attribute*), 30
HashicorpVaultCredential (*class in towerlib.entities.credential*), 20
heartbeat (*towerlib.entities.core.ClusterInstance attribute*), 13
Host (*class in towerlib.entities.host*), 23
host (*towerlib.entities.credential.GenericCredential attribute*), 19
host (*towerlib.entities.job.JobEvent attribute*), 36
host (*towerlib.entities.job.JobRun attribute*), 39
host (*towerlib.entities.job.JobSummary attribute*), 40
host (*towerlib.entities.notification.NotificationEmail attribute*), 53
host_config_key (*towerlib.entities.job.JobTemplate attribute*), 44
host_filter (*towerlib.entities.inventory.Inventory attribute*), 30
host_name (*towerlib.entities.job.JobEvent attribute*), 36

host_name (*towerlib.entities.job.JobSummary attribute*), 41
hostname (*towerlib.entities.core.ClusterInstance attribute*), 14
hostname (*towerlib.entities.core.LicenseInfo attribute*), 17
hostname (*towerlib.entities.instance.Instance attribute*), 26
hosts (*towerlib.entities.group.Group attribute*), 22
hosts (*towerlib.entities.inventory.Inventory attribute*), 31
hosts (*towerlib.towerlib.Tower attribute*), 96
hosts_with_active_failures_count (*towerlib.entities.group.Group attribute*), 22
hosts_with_active_failures_count (*towerlib.entities.inventory.Inventory attribute*), 31

I

id (*towerlib.entities.core.ClusterInstance attribute*), 14
id (*towerlib.entities.core.Entity attribute*), 15
injectors (*towerlib.entities.credential.CredentialType attribute*), 18
input_sources (*towerlib.entities.credential.CredentialType attribute*), 18
inputs (*towerlib.entities.credential.CredentialType attribute*), 18
inputs (*towerlib.entities.credential.GenericCredential attribute*), 19
insights_credential (*towerlib.entities.inventory.Inventory attribute*), 31
insights_system_id (*towerlib.entities.host.Host attribute*), 24
Instance (*class in towerlib.entities.instance*), 25
instance_count (*towerlib.entities.core.LicenseInfo attribute*), 17
instance_filters (*towerlib.entities.inventory_source.InventorySource attribute*), 33
instance_groups (*towerlib.towerlib.Tower attribute*), 96
instance_id (*towerlib.entities.host.Host attribute*), 24
InstanceGroup (*class in towerlib.entities.instance*), 26
instances (*towerlib.entities.instance.InstanceGroup attribute*), 27
instances (*towerlib.towerlib.Tower attribute*), 96
instances_count (*towerlib.entities.instance.InstanceGroup attribute*), 27
InvalidCredential, 99
InvalidCredentialType, 99
InvalidGroup, 99
InvalidHost, 100
InvalidInstanceGroup, 100
InvalidInventory, 100
InvalidInventoryScript, 100
InvalidJobTemplate, 100
InvalidJobType, 100
InvalidOrganization, 100
InvalidPlaybook, 100
InvalidProject, 100
InvalidRole, 100
InvalidSchedule, 100
InvalidTeam, 100
InvalidUser, 100
InvalidUserLevel, 101
InvalidValue, 101
InvalidVariables, 101
InvalidVerbosity, 101
inventories (*towerlib.entities.organization.Organization attribute*), 62
inventories (*towerlib.towerlib.Tower attribute*), 96
inventories_count (*towerlib.entities.organization.Organization attribute*), 63
Inventory (*class in towerlib.entities.inventory*), 27
inventory (*towerlib.entities.group.Group attribute*), 22
inventory (*towerlib.entities.host.Host attribute*), 24
inventory (*towerlib.entities.inventory_source.InventorySource attribute*), 33
inventory (*towerlib.entities.job.AdHocCommandJob attribute*), 35
inventory (*towerlib.entities.job.JobRun attribute*), 39
inventory (*towerlib.entities.job.JobTemplate attribute*), 44
inventory (*towerlib.entities.schedule.Schedule attribute*), 69
inventory_scripts (*towerlib.entities.organization.Organization attribute*), 63
inventory_scripts (*towerlib.towerlib.Tower attribute*), 96
inventory_sources (*towerlib.towerlib.Tower attribute*), 96
inventory_sources_with_failures_count (*towerlib.entities.inventory.Inventory attribute*), 31
InventoryScript (*class in towerlib.entities.inventory_script*), 32
InventorySource (*class in towerlib.entities.inventory_source*), 33
is_changed (*towerlib.entities.job.JobEvent attribute*), 36

is_changed (*towerlib.entities.job.JobSummary attribute*), 41
is_dark (*towerlib.entities.job.JobSummary attribute*), 41
is_failed (*towerlib.entities.job.JobEvent attribute*), 36
is_failed (*towerlib.entities.job.ProjectUpdateJob attribute*), 48
is_failed (*towerlib.entities.job.SystemJob attribute*), 50
is_last_job_failed (*towerlib.entities.job.JobTemplate attribute*), 44
is_last_job_failed (*towerlib.entities.project.Project attribute*), 64
is_last_update_failed (*towerlib.entities.project.Project attribute*), 65
is_ok (*towerlib.entities.job.JobSummary attribute*), 41
is_superuser (*towerlib.entities.user.User attribute*), 78
is_system_auditor (*towerlib.entities.user.User attribute*), 78

J

Job (*class in towerlib.entities.job*), 35
job (*towerlib.entities.job.JobEvent attribute*), 37
job_args (*towerlib.entities.job.ProjectUpdateJob attribute*), 48
job_args (*towerlib.entities.job.SystemJob attribute*), 51
job_cwd (*towerlib.entities.job.ProjectUpdateJob attribute*), 48
job_cwd (*towerlib.entities.job.SystemJob attribute*), 51
job_env (*towerlib.entities.job.ProjectUpdateJob attribute*), 48
job_env (*towerlib.entities.job.SystemJob attribute*), 51
job_events (*towerlib.entities.host.Host attribute*), 25
job_events (*towerlib.entities.job.JobRun attribute*), 39
job_events (*towerlib.towerlib.Tower attribute*), 96
job_explanation (*towerlib.entities.job.ProjectUpdateJob attribute*), 48
job_explanation (*towerlib.entities.job.SystemJob attribute*), 51
job_host_summaries (*towerlib.entities.job.JobRun attribute*), 39
job_tags (*towerlib.entities.job.JobTemplate attribute*), 44
job_tags (*towerlib.entities.schedule.Schedule attribute*), 69
job_template (*towerlib.entities.job.JobRun attribute*), 39
job_templates (*towerlib.entities.project.Project attribute*), 65
job_templates (*towerlib.towerlib.Tower attribute*), 96
job_templates_count (*towerlib.entities.organization.Organization attribute*), 63
job_type (*towerlib.entities.job.JobRun attribute*), 39
job_type (*towerlib.entities.job.JobTemplate attribute*), 44
job_type (*towerlib.entities.job.ProjectUpdateJob attribute*), 48
job_type (*towerlib.entities.job.SystemJob attribute*), 51
job_type (*towerlib.entities.schedule.Schedule attribute*), 69
JobEvent (*class in towerlib.entities.job*), 35
JobRun (*class in towerlib.entities.job*), 38
jobs (*towerlib.entities.instance.Instance attribute*), 26
jobs (*towerlib.towerlib.Tower attribute*), 97
jobs_running (*towerlib.entities.core.ClusterInstance attribute*), 14
jobs_running_count (*towerlib.entities.instance.Instance attribute*), 26
jobs_running_count (*towerlib.entities.instance.InstanceGroup attribute*), 27
JobSummary (*class in towerlib.entities.job*), 40
JobTemplate (*class in towerlib.entities.job*), 41

K

kind (*towerlib.entities.credential.CredentialType attribute*), 18
kind (*towerlib.entities.inventory.Inventory attribute*), 31

L

Label (*class in towerlib.entities.core*), 16
labels (*towerlib.entities.job.JobTemplate attribute*), 44
last_job (*towerlib.entities.host.Host attribute*), 25
last_job (*towerlib.entities.project.Project attribute*), 65
last_job_host_summary (*towerlib.entities.host.Host attribute*), 25
last_job_run (*towerlib.entities.project.Project attribute*), 65
last_job_run_at (*towerlib.entities.job.JobTemplate attribute*), 45
last_job_run_id (*towerlib.entities.job.JobTemplate attribute*), 45
last_login (*towerlib.entities.user.User attribute*), 78
last_name (*towerlib.entities.user.User attribute*), 78
last_update (*towerlib.entities.project.Project attribute*), 65
last_updated (*towerlib.entities.project.Project attribute*), 65

launch() (*towerlib.entities.job.JobTemplate method*), 45
launch_type (*towerlib.entities.job.ProjectUpdateJob attribute*), 48
launch_type (*towerlib.entities.job.SystemJob attribute*), 51
ldap (*towerlib.entities.core.LicenseFeatures attribute*), 16
ldap_dn (*towerlib.entities.user.User attribute*), 78
license_date (*towerlib.entities.core.LicenseInfo attribute*), 17
license_info (*towerlib.entities.core.Config attribute*), 14
license_key (*towerlib.entities.core.LicenseInfo attribute*), 17
license_type (*towerlib.entities.core.LicenseInfo attribute*), 17
LicenseFeatures (*class in towerlib.entities.core*), 16
LicenseInfo (*class in towerlib.entities.core*), 16
limit (*towerlib.entities.job.JobTemplate attribute*), 45
limit (*towerlib.entities.schedule.Schedule attribute*), 69
local_path (*towerlib.entities.job.ProjectUpdateJob attribute*), 48
local_path (*towerlib.entities.project.Project attribute*), 65
local_users (*towerlib.towerlib.Tower attribute*), 97

M

MachineCredential (*class in towerlib.entities.credential*), 20
managed_by_tower (*towerlib.entities.credential.CredentialType attribute*), 18
mattermost_no_verify_ssl (*towerlib.entities.notification.NotificationMatterMost attribute*), 56
mattermost_url (*towerlib.entities.notification.NotificationMatterMost attribute*), 56
message_from (*towerlib.entities.notification.NotificationHipChat attribute*), 55
metadata_url (*towerlib.entities.settings.Saml attribute*), 71
modified_at (*towerlib.entities.core.ClusterInstance attribute*), 14
modified_at (*towerlib.entities.core.Entity attribute*), 15
modified_at (*towerlib.entities.job.JobEvent attribute*), 37
modified_at (*towerlib.entities.job.JobRun attribute*), 39
modified_by (*towerlib.entities.credential.GenericCredential attribute*), 19
modified_by (*towerlib.entities.host.Host attribute*), 25
modified_by (*towerlib.entities.job.JobTemplate attribute*), 45
modified_by (*towerlib.entities.organization.Organization attribute*), 63
module_args (*towerlib.entities.job.AdHocCommandJob attribute*), 35
module_name (*towerlib.entities.job.AdHocCommandJob attribute*), 35
multiple_organizations (*towerlib.entities.core.LicenseFeatures attribute*), 16

N

name (*towerlib.entities.credential.CredentialType attribute*), 19
name (*towerlib.entities.credential.GenericCredential attribute*), 19
name (*towerlib.entities.group.Group attribute*), 22
name (*towerlib.entities.host.Host attribute*), 25
name (*towerlib.entities.instance.InstanceGroup attribute*), 27
name (*towerlib.entities.inventory.Inventory attribute*), 31
name (*towerlib.entities.inventory_script.InventoryScript attribute*), 32
name (*towerlib.entities.inventory_source.InventorySource attribute*), 33
name (*towerlib.entities.job.JobRun attribute*), 39
name (*towerlib.entities.job.JobTemplate attribute*), 45
name (*towerlib.entities.job.ProjectUpdateJob attribute*), 48
name (*towerlib.entities.job.SystemJob attribute*), 51
name (*towerlib.entities.notification.NotificationTemplate attribute*), 58
name (*towerlib.entities.organization.Organization attribute*), 63
name (*towerlib.entities.project.Project attribute*), 65
name (*towerlib.entities.role.Role attribute*), 68
name (*towerlib.entities.schedule.Schedule attribute*), 69
name (*towerlib.entities.team.Team attribute*), 74
next (*towerlib.entities.job.WorkflowNodes attribute*), 52
next_job_run (*towerlib.entities.project.Project attribute*), 65
next_job_run_at (*towerlib.entities.job.JobTemplate attribute*), 45
next_run (*towerlib.entities.schedule.Schedule attribute*), 69
nickname (*towerlib.entities.notification.NotificationIRC attribute*), 55

Notification (*class in towerlib.entities.notification*), 53
notification_configuration (*towerlib.entities.notification.NotificationTemplate attribute*), 58
notification_templates (*towerlib.towerlib.Tower attribute*), 97
notification_type (*towerlib.entities.notification.Notification attribute*), 53
notification_type (*towerlib.entities.notification.NotificationTemplate attribute*), 58
NotificationEmail (*class in towerlib.entities.notification*), 53
NotificationGrafana (*class in towerlib.entities.notification*), 54
NotificationHipChat (*class in towerlib.entities.notification*), 55
NotificationIRC (*class in towerlib.entities.notification*), 55
NotificationMatterMost (*class in towerlib.entities.notification*), 56
NotificationPagerDuty (*class in towerlib.entities.notification*), 56
NotificationRocketChat (*class in towerlib.entities.notification*), 57
notifications_sent (*towerlib.entities.notification.Notification attribute*), 53
NotificationSlack (*class in towerlib.entities.notification*), 57
NotificationTemplate (*class in towerlib.entities.notification*), 57
NotificationTwilio (*class in towerlib.entities.notification*), 58
NotificationWebHook (*class in towerlib.entities.notification*), 59
notify (*towerlib.entities.notification.NotificationHipChat attribute*), 55

O

object_role_names (*towerlib.entities.inventory.Inventory attribute*), 31
object_role_names (*towerlib.entities.organization.Organization attribute*), 63
object_role_names (*towerlib.entities.project.Project attribute*), 65
object_role_names (*towerlib.entities.team.Team attribute*), 74
object_roles (*towerlib.entities.credential.GenericCredential attribute*), 19
object_roles (*towerlib.entities.inventory.Inventory attribute*), 31
object_roles (*towerlib.entities.job.JobTemplate attribute*), 45
object_roles (*towerlib.entities.organization.Organization attribute*), 63
object_roles (*towerlib.entities.project.Project attribute*), 66
object_roles (*towerlib.entities.team.Team attribute*), 75
ObjectRole (*class in towerlib.entities.role*), 67
Organization (*class in towerlib.entities.organization*), 59
organization (*towerlib.entities.credential.GenericCredential attribute*), 20
organization (*towerlib.entities.inventory.Inventory attribute*), 31
organization (*towerlib.entities.notification.NotificationTemplate attribute*), 58
organization (*towerlib.entities.project.Project attribute*), 66
organization (*towerlib.entities.team.Team attribute*), 75
organization_attributes (*towerlib.entities.settings.Saml attribute*), 71
organization_information (*towerlib.entities.settings.Saml attribute*), 71
organization_map (*towerlib.entities.settings.Saml attribute*), 71
organizations (*towerlib.entities.user.User attribute*), 78
organizations (*towerlib.towerlib.Tower attribute*), 97
overwrite (*towerlib.entities.inventory_source.InventorySource attribute*), 33
overwrite_vars (*towerlib.entities.inventory_source.InventorySource attribute*), 33
owner_teams (*towerlib.entities.credential.GenericCredential attribute*), 20
owner_users (*towerlib.entities.credential.GenericCredential attribute*), 20

P

parent (*towerlib.entities.job.JobEvent attribute*), 37
parent_uuid (*towerlib.entities.job.JobEvent attribute*), 37

password (*towerlib.entities.credential.MachineCredential attribute*), 20
password (*towerlib.entities.notification.NotificationEmail attribute*), 53
password (*towerlib.entities.notification.NotificationIRC attribute*), 56
password (*towerlib.entities.user.User attribute*), 78
pending_deletion (*towerlib.entities.inventory.Inventory attribute*), 31
percent_capacity_remaining (*towerlib.entities.core.ClusterInstance attribute*), 14
percent_capacity_remaining (*towerlib.entities.instance.Instance attribute*), 26
percent_capacity_remaining (*towerlib.entities.instance.InstanceGroup attribute*), 27
PermissionNotFound, 101
play (*towerlib.entities.job.JobEvent attribute*), 37
playbook (*towerlib.entities.job.JobEvent attribute*), 37
playbook (*towerlib.entities.job.JobTemplate attribute*), 45
playbooks (*towerlib.entities.project.Project attribute*), 66
port (*towerlib.entities.notification.NotificationEmail attribute*), 54
port (*towerlib.entities.notification.NotificationIRC attribute*), 56
previous (*towerlib.entities.job.WorkflowNodes attribute*), 52
processed_count (*towerlib.entities.job.JobSummary attribute*), 41
Project (*class in towerlib.entities.project*), 64
project (*towerlib.entities.credential.GenericCredential attribute*), 20
project (*towerlib.entities.job.JobRun attribute*), 39
project (*towerlib.entities.job.JobTemplate attribute*), 45
project (*towerlib.entities.job.ProjectUpdateJob attribute*), 49
project_base_dir (*towerlib.entities.core.Config attribute*), 14
project_local_paths (*towerlib.entities.core.Config attribute*), 14
project_updates (*towerlib.entities.project.Project attribute*), 66
project_updates (*towerlib.towerlib.Tower attribute*), 97
projects (*towerlib.entities.organization.Organization attribute*), 63
projects (*towerlib.entities.role.Role attribute*), 68
projects (*towerlib.entities.team.Team attribute*), 75
projects (*towerlib.entities.user.User attribute*), 78
projects (*towerlib.towerlib.Tower attribute*), 97
projects_count (*towerlib.entities.organization.Organization attribute*), 63
ProjectUpdateJob (*class in towerlib.entities.job*), 47

R

rebranding (*towerlib.entities.core.LicenseFeatures attribute*), 16
recent_jobs (*towerlib.entities.host.Host attribute*), 25
recent_jobs (*towerlib.entities.job.JobTemplate attribute*), 46
recent_notifications (*towerlib.entities.notification.NotificationTemplate attribute*), 58
recipients (*towerlib.entities.notification.Notification attribute*), 53
recipients (*towerlib.entities.notification.NotificationEmail attribute*), 54
recurrence_rule (*towerlib.entities.schedule.Schedule attribute*), 70
remove_credential_permission_admin() (*towerlib.entities.team.Team method*), 75
remove_credential_permission_use() (*towerlib.entities.team.Team method*), 75
remove_host_by_name() (*towerlib.entities.group.Group method*), 22
remove_inventory_permission_ad_hoc() (*towerlib.entities.team.Team method*), 75
remove_inventory_permission_admin() (*towerlib.entities.team.Team method*), 75
remove_inventory_permission_update() (*towerlib.entities.team.Team method*), 75
remove_inventory_permission_use() (*towerlib.entities.team.Team method*), 76
remove_job_template_permission_admin() (*towerlib.entities.team.Team method*), 76
remove_job_template_permission_execute() (*towerlib.entities.team.Team method*), 76
remove_organization_role_by_name() (*towerlib.entities.team.Team method*), 76
remove_project_permission_admin() (*towerlib.entities.team.Team method*), 76
remove_project_permission_update() (*towerlib.entities.team.Team method*), 76
remove_project_permission_use() (*towerlib.entities.team.Team method*), 76
remove_user_as_admin() (*towerlib.entities.team.Team method*), 76
remove_user_as_member() (*towerlib.entities.team.Team method*), 76

result_stdout (*towerlib.entities.job.JobRun attribute*), 40
result_stdout (*towerlib.entities.job.ProjectUpdateJob attribute*), 49
result_stdout (*towerlib.entities.job.SystemJob attribute*), 51
result_traceback (*towerlib.entities.job.JobRun attribute*), 40
result_traceback (*towerlib.entities.job.ProjectUpdateJob attribute*), 49
result_traceback (*towerlib.entities.job.SystemJob attribute*), 51
results (*towerlib.entities.job.WorkflowNodes attribute*), 52
rocketchat_no_verify_ssl (*towerlib.entities.notification.NotificationRocketChat attribute*), 57
rocketchat_url (*towerlib.entities.notification.NotificationRocketChat attribute*), 57
Role (*class in towerlib.entities.role*), 67
role (*towerlib.entities.job.JobEvent attribute*), 37
roles (*towerlib.entities.team.Team attribute*), 76
roles (*towerlib.entities.user.User attribute*), 79
roles (*towerlib.towerlib.Tower attribute*), 97
rooms (*towerlib.entities.notification.NotificationHipChat attribute*), 55

S

Saml (*class in towerlib.entities.settings*), 70
saml (*towerlib.entities.settings.Settings attribute*), 72
Schedule (*class in towerlib.entities.schedule*), 68
schedules (*towerlib.entities.job.JobTemplate attribute*), 46
schedules (*towerlib.towerlib.Tower attribute*), 97
scm_branch (*towerlib.entities.job.ProjectUpdateJob attribute*), 49
scm_branch (*towerlib.entities.project.Project attribute*), 66
scm_branch (*towerlib.entities.schedule.Schedule attribute*), 70
scm_clean (*towerlib.entities.job.ProjectUpdateJob attribute*), 49
scm_clean (*towerlib.entities.project.Project attribute*), 66
scm_delete_on_next_update (*towerlib.entities.project.Project attribute*), 66
scm_delete_on_update (*towerlib.entities.job.ProjectUpdateJob attribute*), 49
scm_delete_on_update (*towerlib.entities.project.Project attribute*), 66
scm_revision (*towerlib.entities.project.Project attribute*), 66
scm_type (*towerlib.entities.job.ProjectUpdateJob attribute*), 49
scm_type (*towerlib.entities.project.Project attribute*), 66
scm_update_cache_timeout (*towerlib.entities.project.Project attribute*), 67
scm_update_on_launch (*towerlib.entities.project.Project attribute*), 67
scm_url (*towerlib.entities.job.ProjectUpdateJob attribute*), 49
scm_url (*towerlib.entities.project.Project attribute*), 67
script (*towerlib.entities.inventory_script.InventoryScript attribute*), 32
secret_key (*towerlib.entities.credential.AmazonWebServicesCredential attribute*), 18
security_config (*towerlib.entities.settings.Saml attribute*), 71
sender (*towerlib.entities.notification.NotificationEmail attribute*), 54
server (*towerlib.entities.notification.NotificationIRC attribute*), 56
service_key (*towerlib.entities.notification.NotificationPagerDuty attribute*), 56
set_inventory() (*towerlib.entities.job.JobTemplate method*), 46
Settings (*class in towerlib.entities.settings*), 72
settings (*towerlib.towerlib.Tower attribute*), 97
skip_tags (*towerlib.entities.job.JobTemplate attribute*), 46
skip_tags (*towerlib.entities.schedule.Schedule attribute*), 70
skipped_count (*towerlib.entities.job.JobSummary attribute*), 41
source (*towerlib.entities.inventory_source.InventorySource attribute*), 33
source_path (*towerlib.entities.inventory_source.InventorySource attribute*), 33
source_project (*towerlib.entities.inventory_source.InventorySource attribute*), 33
source_regions (*towerlib.entities.inventory_source.InventorySource attribute*), 34
source_script (*towerlib.entities.inventory_source.InventorySource attribute*), 34
source_vars (*towerlib.entities.inventory_source.InventorySource attribute*), 34
sources (*towerlib.entities.inventory.Inventory at-*

tribute), 32
sp_entity_id (*towerlib.entities.settings.Saml attribute*), 71
sp_extra (*towerlib.entities.settings.Saml attribute*), 71
sp_private_key (*towerlib.entities.settings.Saml attribute*), 71
sp_public_cert (*towerlib.entities.settings.Saml attribute*), 72
start_at_task (*towerlib.entities.job.JobTemplate attribute*), 46
start_line (*towerlib.entities.job.JobEvent attribute*), 37
started_at (*towerlib.entities.job.ProjectUpdateJob attribute*), 49
started_at (*towerlib.entities.job.SystemJob attribute*), 51
status (*towerlib.entities.job.JobRun attribute*), 40
status (*towerlib.entities.job.JobTemplate attribute*), 46
status (*towerlib.entities.job.ProjectUpdateJob attribute*), 49
status (*towerlib.entities.job.SystemJob attribute*), 52
status (*towerlib.entities.notification.Notification attribute*), 53
status (*towerlib.entities.project.Project attribute*), 67
stdout (*towerlib.entities.job.JobEvent attribute*), 37
stdout (*towerlib.entities.job.JobRun attribute*), 40
stdout (*towerlib.entities.job.ProjectUpdateJob attribute*), 50
subdomain (*towerlib.entities.notification.NotificationPagerDuty attribute*), 57
subject (*towerlib.entities.notification.Notification attribute*), 53
subscription_name (*towerlib.entities.core.LicenseInfo attribute*), 17
summary_fields (*towerlib.entities.job.JobRun attribute*), 40
summary_fields (*towerlib.entities.job.JobSummary attribute*), 41
summary_fields (*towerlib.entities.job.JobTemplate attribute*), 46
summary_fields (*towerlib.entities.job.ProjectUpdateJob attribute*), 50
summary_fields (*towerlib.entities.job.SystemJob attribute*), 52
summary_fields (*towerlib.entities.role.Role attribute*), 68
support_contact (*towerlib.entities.settings.Saml attribute*), 72
survey_enabled (*towerlib.entities.job.JobTemplate attribute*), 46
survey_spec (*towerlib.entities.job.JobTemplate attribute*), 46
surveys (*towerlib.entities.core.LicenseFeatures attribute*), 16
system_jobs (*towerlib.towerlib.Tower attribute*), 98
system_tracking (*towerlib.entities.core.LicenseFeatures attribute*), 16
SystemJob (*class in towerlib.entities.job*), 50

T

targets (*towerlib.entities.notification.NotificationIRC attribute*), 56
task (*towerlib.entities.job.JobEvent attribute*), 37
Team (*class in towerlib.entities.team*), 72
team (*towerlib.entities.role.ObjectRole attribute*), 67
team_attributes (*towerlib.entities.settings.Saml attribute*), 72
team_map (*towerlib.entities.settings.Saml attribute*), 72
teams (*towerlib.entities.organization.Organization attribute*), 63
teams (*towerlib.entities.role.Role attribute*), 68
teams (*towerlib.entities.user.User attribute*), 79
teams (*towerlib.towerlib.Tower attribute*), 98
teams_count (*towerlib.entities.organization.Organization attribute*), 64
technical_contact (*towerlib.entities.settings.Saml attribute*), 72
time_remaining (*towerlib.entities.core.LicenseInfo attribute*), 17
time_zone (*towerlib.entities.core.Config attribute*), 14
timeout (*towerlib.entities.inventory_source.InventorySource attribute*), 34
timeout (*towerlib.entities.job.JobTemplate attribute*), 47
timeout (*towerlib.entities.job.ProjectUpdateJob attribute*), 50
timeout (*towerlib.entities.notification.NotificationEmail attribute*), 54
timeout (*towerlib.entities.project.Project attribute*), 67
timezone (*towerlib.entities.schedule.Schedule attribute*), 70
to_numbers (*towerlib.entities.notification.NotificationTwilio attribute*), 58
token (*towerlib.entities.credential.HashicorpVaultCredential attribute*), 20
token (*towerlib.entities.notification.NotificationHipChat attribute*), 55
token (*towerlib.entities.notification.NotificationPagerDuty attribute*), 57
token (*towerlib.entities.notification.NotificationSlack attribute*), 57
total_groups_count (*towerlib.entities.group.Group attribute*), 22
total_groups_count (*towerlib.entities.inventory.Inventory attribute*),

32
total_hosts_count (*towerlib.entities.group.Group attribute*), 23
total_hosts_count (*towerlib.entities.inventory.Inventory attribute*), 32
total_inventory_sources_count (*towerlib.entities.inventory.Inventory attribute*), 32
Tower (*class in towerlib.towerlib*), 79
tower_credential_types (*towerlib.towerlib.Tower attribute*), 98
towerlib (*module*), 101
towerlib.entities (*module*), 79
towerlib.entities.core (*module*), 13
towerlib.entities.credential (*module*), 17
towerlib.entities.group (*module*), 21
towerlib.entities.host (*module*), 23
towerlib.entities.instance (*module*), 25
towerlib.entities.inventory (*module*), 27
towerlib.entities.inventory_script (*module*), 32
towerlib.entities.inventory_source (*module*), 33
towerlib.entities.job (*module*), 34
towerlib.entities.notification (*module*), 53
towerlib.entities.organization (*module*), 59
towerlib.entities.project (*module*), 64
towerlib.entities.role (*module*), 67
towerlib.entities.schedule (*module*), 68
towerlib.entities.settings (*module*), 70
towerlib.entities.team (*module*), 72
towerlib.entities.user (*module*), 77
towerlib.towerlib (*module*), 79
towerlib.towerlibexceptions (*module*), 99
trial (*towerlib.entities.core.LicenseInfo attribute*), 17
type (*towerlib.entities.core.Entity attribute*), 15

U

unified_job_template (*towerlib.entities.job.JobRun attribute*), 40
unified_job_template (*towerlib.entities.schedule.Schedule attribute*), 70
unified_job_templates (*towerlib.towerlib.Tower attribute*), 98
unified_jobs (*towerlib.towerlib.Tower attribute*), 98
until (*towerlib.entities.schedule.Schedule attribute*), 70
update() (*towerlib.entities.project.Project method*), 67
update_all_organization_projects() (*towerlib.towerlib.Tower method*), 98
update_cache_timeout (*towerlib.entities.inventory_source.InventorySource attribute*), 34
update_on_launch (*towerlib.entities.inventory_source.InventorySource attribute*), 34
update_on_project_update (*towerlib.entities.inventory_source.InventorySource attribute*), 34
update_organization_project_by_name() (*towerlib.towerlib.Tower method*), 98
update_organization_projects_by_branch_name() (*towerlib.towerlib.Tower method*), 98
update_organization_projects_by_scm_url() (*towerlib.towerlib.Tower method*), 99
update_project_by_id() (*towerlib.towerlib.Tower method*), 99
url (*towerlib.entities.core.Entity attribute*), 15
url (*towerlib.entities.notification.NotificationWebHook attribute*), 59
url (*towerlib.entities.settings.Saml attribute*), 72
use_fact_cache (*towerlib.entities.job.JobTemplate attribute*), 47
use_ssl (*towerlib.entities.notification.NotificationEmail attribute*), 54
use_ssl (*towerlib.entities.notification.NotificationIRC attribute*), 56
use_tls (*towerlib.entities.notification.NotificationEmail attribute*), 54
User (*class in towerlib.entities.user*), 77
username (*towerlib.entities.credential.MachineCredential attribute*), 20
username (*towerlib.entities.notification.NotificationEmail attribute*), 54
username (*towerlib.entities.user.User attribute*), 79
users (*towerlib.entities.organization.Organization attribute*), 64
users (*towerlib.entities.role.Role attribute*), 68
users (*towerlib.entities.team.Team attribute*), 77
users (*towerlib.towerlib.Tower attribute*), 99
users_count (*towerlib.entities.organization.Organization attribute*), 64
uuid (*towerlib.entities.core.ClusterInstance attribute*), 14
uuid (*towerlib.entities.instance.Instance attribute*), 26
uuid (*towerlib.entities.job.JobEvent attribute*), 38

V

valid_key (*towerlib.entities.core.LicenseInfo attribute*), 17
validate_characters() (*in module towerlib.entities.core*), 17

validate_json() (*in module towerlib.entities.core*), 17
validate_max_length() (*in module towerlib.entities.core*), 17
validate_range() (*in module towerlib.entities.core*), 17
variables (*towerlib.entities.group.Group attribute*), 23
variables (*towerlib.entities.host.Host attribute*), 25
variables (*towerlib.entities.inventory.Inventory attribute*), 32
vault_address (*towerlib.entities.credential.HashicorpVaultCredential attribute*), 20
vault_credential (*towerlib.entities.job.JobTemplate attribute*), 47
verbosity (*towerlib.entities.inventory_source.InventorySource attribute*), 34
verbosity (*towerlib.entities.job.JobEvent attribute*), 38
verbosity (*towerlib.entities.job.JobTemplate attribute*), 47
verbosity (*towerlib.entities.schedule.Schedule attribute*), 70
version (*towerlib.entities.core.ClusterInstance attribute*), 14
version (*towerlib.entities.core.Config attribute*), 14
version (*towerlib.entities.instance.Instance attribute*), 26

W

workflow_job_templates (*towerlib.towerlib.Tower attribute*), 99
workflow_jobs (*towerlib.towerlib.Tower attribute*), 99
workflow_nodes (*towerlib.entities.job.WorkflowJobRun attribute*), 52
WorkflowJobRun (*class in towerlib.entities.job*), 52
WorkflowNodes (*class in towerlib.entities.job*), 52
workflows (*towerlib.entities.core.LicenseFeatures attribute*), 16