
Robottelo Documentation

Release 0.0.1

Og Maciel <omaciel@redhat.com>

Aug 13, 2020

Contents

1 Quickstart	3
1.1 Robottelo on Docker	4
2 Running the Tests	5
2.1 Initial Configuration	5
2.2 Running the UI Tests in headless mode	6
2.3 Testing With Pytest	6
2.4 Testing With Unittest	7
2.5 Testing With Nose	7
2.6 Running UI Tests On a Docker Browser	8
2.7 Running UI Tests On SauceLabs	8
3 Miscellany	11
3.1 Committing	11
3.2 Code Standards	13
3.3 Reviewing PRs	15
3.4 Features	15
3.5 API Reference	22
Python Module Index	191
Index	193

Robottelo is a test suite which exercises [The Foreman](#). All tests are automated, suited for use in a continuous integration environment, and [data driven](#). There are three types of tests:

- UI tests, which rely on Selenium's [WebDriver](#).
- CLI tests, which rely on [Paramiko](#).
- API tests, which rely on [Requests](#).

Contents

- *Robottelo*
- *Quickstart*
 - *Robottelo on Docker*
- *Running the Tests*
 - *Initial Configuration*
 - * *Using environment variables*
 - *Running the UI Tests in headless mode*
 - *Testing With Pytest*
 - *Testing With Unittest*
 - *Testing With Nose*
 - *Running UI Tests On a Docker Browser*
 - *Running UI Tests On SauceLabs*
- *Miscellany*

The following is only a brief setup guide for [Robottelo](#). The section on *Running the Tests* provides a more comprehensive guide to using Robottelo.

Robottelo requires SSH access to the Satellite 6 system under test, and this SSH access is implemented by Paramiko. Install the headers for the following to ensure that Paramiko's dependencies build correctly:

- OpenSSL
- Python development headers
- libffi

On Fedora, you can install these with the following command:

For python2.x:

```
dnf install -y gcc git libffi-devel openssl-devel python2-devel \
    redhat-rpm-config libcurl-devel libxml2-devel
```

For python3.x:

```
dnf install -y gcc git libffi-devel openssl-devel python3-devel \
    redhat-rpm-config libcurl-devel libxml2-devel
```

On Red Hat Enterprise Linux 7, you can install these with the following command:

```
yum install -y gcc git libffi-devel openssl-devel python-devel \
    redhat-rpm-config libcurl-devel libxml2-devel
```

For more information, see [Paramiko: Installing](#).

Get the source code and install dependencies:

```
$ git clone git://github.com/SatelliteQE/robottelo.git
$ export PYCURL_SSL_LIBRARY=<ssl library>
$ pip install -r requirements.txt
```

Notes:

- For python 2.7, run `pip install configparser` for Satellite 6.2
- To determine ssl library, check <http://pycurl.io/docs/latest/install.html#ssl>

That's it! You can now go ahead and start testing The Foreman. However, there are a few other things you may wish to do before continuing:

1. You may want to install development tools (such as `gcc`) for your OS. If running Fedora or Red Hat Enterprise Linux, execute `yum groupinstall "Development Tools"`. Make sure to use `dnf` instead of `yum` if `dnf` is available on your system.
2. You may wish to install the optional dependencies listed in `requirements-optional.txt`. (Use `pip`, as shown above.) They are required for tasks like working with certificates, running the internal robottelo test suite and checking code quality with `pre-commit`.

1.1 Robottelo on Docker

Robottelo is also available on [dockerhub](#):

```
$ docker pull satelliteteq/robottelo
```

It also can be built locally using the Dockerfile, in the main directory.:

```
$ docker build -t robottelo .
```

In order to run tests, you will need to mount your `robottelo.properties` file.:

```
$ docker run -v {path to robottelo dir}/robottelo.properties:/robottelo/robottelo.  
↳properties satelliteteq/robottelo <test command>
```

You can also mount the entire robottelo directory to include the properties file and any new tests you have written.:

```
$ docker run -it -v {path to robottelo dir}:/robottelo satelliteteq/robottelo /bin/bash
```

Notes:

- CLI tests run easiest if you include the root credentials in `robottelo.properties`
- UI tests should be configured to run through your SauceLabs account.

Running the Tests

Before running any tests, you must create a configuration file:

```
$ cp robottelo.properties.sample robottelo.properties
$ vi robottelo.properties
```

That done, you can run tests using make:

```
$ make test-robottelo
$ make test-docstrings
$ make test-foreman-api
$ make test-foreman-cli
$ make test-foreman-ui
$ make test-foreman-smoke
```

Robottelo provides two test suites, one for testing Robottelo itself and another for testing Foreman/Satellite 6. Robottelo's tests are under the tests/robottelo directory and the Foreman/Satellite 6 tests are under the tests/foreman directory.

If you want to run tests without the aid of make, you can do that with either `pytest`, `unittest` or `nose`. Just specify the path for the test suite you want to run:

```
$ pytest tests/robotello
$ pytest tests/foreman
$ python -m unittest discover -s tests/robottelo -t .
$ python -m unittest discover -s tests/foreman -t .
$ nosetests tests/robottelo
$ nosetests tests/foreman
```

The following sections discuss, in detail, how to update the configuration file and run tests directly.

2.1 Initial Configuration

To configure Robottelo, create a file named `robottelo.properties`. You can use the `robottelo.properties.sample` file as a starting point. Then, edit the configuration file so that at least the following attributes

are set:

```
[server]
hostname=[FULLY QUALIFIED DOMAIN NAME OR IP ADDRESS]
ssh_key=[PATH TO YOUR SSH KEY]

[bugzilla]
api_key=sdfsdg654g8df4gdf6g4df8g468dfg
```

Note that you only need to configure the SSH key if you want to run CLI tests. There are other settings to configure what web browser to use for UI tests and even configuration to run the automation using [SauceLabs](#). For more information about what web browsers you can use, check Selenium's [WebDriver](#) documentation.

2.1.1 Using environment variables

Each of the sections in the `robottelo.properties` file can be mapped to an environment variable prefixed with `ROBOTTELO_` so for example if you want to override the `server.hostname` without changing the properties file you can do:

```
$ export ROBOTTELO_SERVER_HOSTNAME=other.hostname.com
```

The envvars follows the format `ROBOTTELO_{SECTION}_{VALUE}` all uppercase, more examples:

```
$ export ROBOTTELO_SERVER_SSH_KEY=path/to/your/key
$ export ROBOTTELO_BUGZILLA_API_KEY=sdfsdg654g8df4gdf6g4df8g468dfg
```

2.2 Running the UI Tests in headless mode

You can run browser for UI tests in headless mode by setting `browser` option in `robottelo.properties` file. Currently it is supported only for chrome

```
browseroptions=headless
```

2.3 Testing With Pytest

To run all tests:

```
$ pytest
```

It is possible to run a specific subset of tests:

```
$ pytest test_case.py
$ pytest test_case.py::TestClass
$ pytest test_case.py::TestClass::test_case_name
```

To get more verbose output, or run multiple tests:

```
$ pytest tests/ -v
$ pytest tests/robottelo/test_decorators.py \
    tests/robottelo/test_cli.py
```

To test The Foreman's API, CLI or UI, use the following commands respectively:

```
$ pytest tests/foreman/api/
$ pytest tests/foreman/cli/
$ pytest tests/foreman/ui/
```

To collect from three directories in one run:

```
$ pytest tests/foreman/{cli,api,ui}/test_host.py
```

To search in testcase names, in this case it will run just negative tests:

```
$ pytest tests/foreman/cli/test_host.py -k negative
```

To run tests in several threads, in this case 4:

```
$ pytest tests/foreman/cli/test_host.py -n 4
```

For more information about Python's `pytest` module, read the documentation.

2.4 Testing With Unittest

To run all tests:

```
$ python -m unittest discover \
  --start-directory tests/ \
  --top-level-directory .
```

It is possible to run a specific subset of tests:

```
$ python -m unittest tests.robottelo.test_decorators
$ python -m unittest tests.robottelo.test_decorators.DataDecoratorTestCase
$ python -m unittest tests.robottelo.test_decorators.DataDecoratorTestCase.test_data_
↪decorator_smoke
```

To get more verbose output, or run multiple tests:

```
$ python -m unittest discover -s tests/ -t . -v
$ python -m unittest \
  tests.robottelo.test_decorators \
  tests.robottelo.test_cli
```

To test The Foreman's API, CLI or UI, use the following commands respectively:

```
$ python -m unittest discover -s tests/foreman/api/
$ python -m unittest discover -s tests/foreman/cli/
$ python -m unittest discover -s tests/foreman/ui/
```

For more information about Python's `unittest` module, read the documentation.

2.5 Testing With Nose

You must have `nose` installed to execute the `nosetests` command.

To run all tests:

```
$ nosetests
```

It is possible to run a specific subset of tests:

```
$ nosetests tests.robottelo.test_decorators
$ nosetests tests.robottelo.test_decorators:DataDecoratorTestCase
$ nosetests tests.robottelo.test_decorators:DataDecoratorTestCase.test_data_decorator_
→smoke
```

To get more verbose output, or run multiple tests:

```
$ nosetests -v
$ nosetests tests.robottelo.test_decorators tests.robottelo.test_cli
```

To test The Foreman's API, CLI or UI, use the following commands respectively:

```
$ nosetests tests.foreman.api
$ nosetests tests.foreman.cli
$ nosetests tests.foreman.ui
```

Many of the existing tests use `subTest` to allow for a more data-driven methodology. In order to run a specific test you need to override the way `nosetests` discovers test names. For instance, if you wanted to run only the `test_positive_create_1` data-driven tests for the `foreman.cli.test_org` module:

```
$ nosetests -m test_positive_create_1 tests.foreman.cli.test_org
```

2.6 Running UI Tests On a Docker Browser

It is possible to run UI tests within a docker container. To do this:

- Install `docker`. It is provided by the `docker` package on Fedora and Red Hat. Be aware that the package may call `docker-io` on old OS releases.
- Make sure that `docker` is up and running and the user that will run `robottelo` has permission to run `docker` commands. For more information check the `docker` installation guide <https://docs.docker.com/engine/installation/>.
- Pull the `selenium/standalone-firefox` image
- Set `browser=docker` at the `[robottelo]` section in the configuration file `robottelo.properties`.

Once you've performed these steps, UI tests will no longer launch a web browser on your system. Instead, UI tests launch a web browser within a docker container.

2.7 Running UI Tests On SauceLabs

It is possible to run UI tests on SauceLabs. To do this:

- Set `browser=saucelabs` at the `[robottelo]` section in the configuration file `robottelo.properties`.
- Select the browser type by setting `webdriver` at the `[robottelo]` section in the configuration file. Valid values are `firefox`, `chrome` and `ie`.
- Fill `saucelabs_user` and `saucelabs_key` at the `[robottelo]` section in the configuration file with your Sauce OnDemand credentials.

- If the machine where Satellite 6 is installed is on a VPN or behind a firewall make sure to have SauceConnect running.
- Optional: install `sauceclient` python package if you want robottelo to report test success or failure back to SauceLabs.

3.1 Committing

Every open source project lives from the generous help by contributors that sacrifice their time and **Robottelo** is no different.

To make participation as pleasant as possible, this project adheres to the [Code of Conduct](#) by the Python Software Foundation.

If you have something great but aren't sure whether it adheres, or even can adhere, to the rules below: please submit a pull request anyway! In the best case, we can mold it into something; in the worst case, the pull request gets politely closed. There's absolutely nothing to fear.

Thank you for considering contributing to Robottelo! If you have any question or concerns, feel free to reach out to the team. We can be found on the [#robottelo](#) channel on [freenode](#).

3.1.1 Before you commit

Ensure your code follows the guidelines in the **Robottelo** *code standards*.

All modules, classes, and functions should have well written docstrings.

See also:

- [testimony](#)
- [sphinx](#)

Don't *ever* break backward compatibility. If it ever has to happen for higher reasons, Robottelo will follow the proven [procedures](#) of the Twisted project.

Always add tests and docs for your code. This is a hard rule; patches with missing tests or documentation won't be merged. If a feature is not tested or documented, it doesn't exist.

In order to ensure you are able to pass the Travis CI build, it is recommended that you run the following commands in the base of your Robottelo directory.:

```
$ flake8 .
$ make test-docstrings
$ make test-robottelo
```

`flake8` will ensure that the changes you made are not in violation of PEP8 standards. If the command gives no output, then you have passed. If not, then address any corrections recommended.

`test-docstrings` will check if there are parts of the code base that are missing docstrings.

`test-robottelo` will run all the tests created to ensure Robottelo is working as expected. If you added a test, it will be run now.

3.1.2 Performing the commit

Proper commit messages

- The title should be a brief summary of the changes.
- The body of the message should effectively explain the changes.
- [This link](#) has much more information on proper commit messages.
- [Close an issue](#) by stating “Fixes #XXXX” where XXXX is the issue number.

Fetch and rebase from upstream/master

Push the commit to your forked repository

3.1.3 Initiating a Pull Request

Navigate to your forked repository on GitHub and click the Pull Request button.

The title and message should auto-populate from your commit. If not, then recreate them now.

Submit the request!

Did you add or change a test? Include the results in a comment.

Closely monitor the discussion on GitHub to address any questions or suggestions.

Never merge your own pull request. Other members of the team are responsible for that action.

3.1.4 Making Changes

If the code review process identifies something that needs to be changed, perform the change locally.

Use `commit --amend` or `squash` the change commit down into the previous.

Finally, push again¹ and continue monitoring the discussion.

3.1.5 Cleaning up

If you created a branch, in your forked repository, you may merge it now.

¹ git may kick back that your GitHub repo is ahead of your current commit. Just use the `--force` to push it anyway.

3.2 Code Standards

3.2.1 General

In order to provide a code base that is easy to maintain, as well as easy to contribute to, **Robottelo** has adopted a set of code standards that all contributors are held to. Violations to our strictly held standards will result in a rejected pull request until all violations have been resolved. While not adhering to our recommended standards isn't a show stopper, keeping to them will help our code base to stay great!

3.2.2 Strictly Held

Linting

- All code will be linted to [PEP8](#) standards using `flake8`.
- In the root of the **Robottelo** directory, run `flake8 .`
- If `flake8` returns errors, make corrections before submitting a pull request.
- pre-commit configuration is available, and its use is strongly encouraged in local development.

Docstrings

- Every class, method and function will have a [properly formatted](#) docstring.
- Docstring will also contain [testimony](#) and [sphinx](#) directives, as appropriate.
 - testimony docstrings are specific to foreman tests, so every test in `tests/foreman/*` should have the testimony tags.
- In the root of the **Robottelo** directory, run `make test-docstrings` to ensure you did not miss adding a docstring.

Strings

- Use string methods instead of the string module.
- Use the `.format()` method to [properly format](#) strings:

```
formatted_string = 'I {0} a {1} string.'.format('am', 'formatted')
```

Naming

- `module_name`
- `method_name`
- `function_name`
- `instance_variable_name`
- `local_variable_name`
- `function_parameter_name`
- `ClassName`
- `ExceptionName`
- `GLOBAL_CONSTANT_NAME`
- `_private_method`
- `_private_variable`

- `_PrivateClass`

Style

- Absolutely no semi colons.
- Lines are not to exceed 99 characters in length.
- Code will be indented with **4 spaces** instead of tabs.

3.2.3 Recommended

Importing

- Import modules in alphabetical order.
- Each non-module import should be on their own line.
- Import order is:
 - standard library
 - third-party
 - application-specific
- Put a blank line between groups of imports.

General

- Avoid global variables.
- Use `.join()` for string concatenation.
- Handle data aggregation inside of functions when able.

Style

- Use single quotes instead of double quotes whenever possible. Single quotes are less visually noisy, and they are easier to type.
- One statement per line.

3.2.4 Read More!

[Python Style Guide](#)

[pre-commit Tutorial](#)

[Code Like a Pythonista](#)

3.2.5 Todo

Compile a list of standards used by the **Robottelo** team

Categorize each standard into how strictly they are enforced

3.3 Reviewing PRs

Ensure commits are squashed.

Thoroughly review each change for errors or departures from the **Robottelo** *code standards*.

If you note something that should be changed, or want clarification, add a comment to the line of code or the PR as a whole.

If tests were added, ensure the PR includes the results or simply a message stating “*All tests passed.*” or “*Unable to run tests at this time due to _____.*”

Don’t merge unless there have been two ACKs.

Don’t merge unless the PR passes the Travis CI build.

Once all the above criteria have been met, and the author has completed all changes, then you may merge.

3.3.1 ToDo

Elaborate on the process.

3.4 Features

This is Robottelo Features documentation. It contains a high level explanation about its capabilities.

3.4.1 Commands

This section explains Robottelo Management Commands.

Contents

- *Commands*
 - *Modules*
 - *requisites*
 - *shell*
 - *ui browse*

Modules

Robottelo uses *manage* and *click* for its management commands, the commands are accessible via console using the `$ manage`. To see a list of all available commands run `$ manage --help`.

The command manager specification is located at *robottelo/manage.yml* and it is easy to add new commands to framework by creating new functions or click commands and referring on *manage.yml* file.

There are commands to open an interactive shell, interact with UI browser, API and other utilities.

requisites

To play with the management commands you need to install robottelo's additional requirements which includes the *manage* library and also is recommended to have iPython installed:

```
$ cd robottelo
$ make pyc-clean
$ pip install ipython
$ pip install -r -U requirements-optional.txt
```

shell

The Interactive Shell is useful to test robottelo's functions and entities, you can change code and all changes are automatically reloaded into shell so no need to restart the shell or reload the modules.

If ipython is installed your shell will be opened using ipython but it is also possible to use *-bpython*, *-ptpython* or bare *-python* as console.

To check the shell options run:

```
$ manage shell --help
```

And to open the shell:

```
$ manage shell
```

Assuming you have ipython installed you will see a console like:

```
(robottelo_env)[you@host robottelo]$ manage shell
Python 2.7.11 (default, Jul  8 2016, 19:45:00)
Type "copyright", "credits" or "license" for more information.

IPython 5.0.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

Welcome to Robottelo Interactive shell
  Auto imported: ['rt', 'nailgun', 'settings', 'robottelo', 'ng', 'entities',
->'locators']

In [1]: rt.ssh.command('uname -r')
2016-09-16 13:54:57 - robottelo.ssh - DEBUG - Connected to [foreman-server.com]
Out[1]: SSHCommandResult(stdout=['3.10.0-327.el7.x86_64', ''], stderr='', return_
->code=0, output_format=None)
In [2]: exit
```

This is the Robottelo's interactive shell welcome screen and you can see some most commonly used objects are *auto_imported* saving you time.

Also the *settings* object is loaded and configured, so you have a ready to use environment to play with robottelo features.

ui browse

In the subgroup *ui* you can find the *browse* command which opens the same interactive shell but it also opens a new browser instance and gives you the context to play with this.

The interaction with the *ui* browser is done trough the *session* object, and the opened browser uses the configuration from your *robottelo.properties* file.

Open a new REPL connected to a browser session:

```
(robottelo_env)[you@host robottelo]$ manage ui browse
2016-09-16 14:00:42 - robottelo.ui.browser - DEBUG - newSession: {
  ↳'desiredCapabilities': {'platform': 'ANY', 'browserName': 'chrome', 'version': '',
  ↳'chromeOptions': {'args': [], 'extensions': []}, 'javascriptEnabled': True}}

Welcome to Robottelo Interactive shell
  Auto imported: ['rt', 'nailgun', 'settings', 'robottelo', 'ng', 'entities', 'host
  ↳', 'session', 'current_browser', 'locators', 'ui_factory', 'api_factory', 'browser']

In [1]: session.browser
Out[1]: <robottelo.ui.browser.Chrome (session="0968e34f29e2c3208554ada58023fa4f")>

In [2]: session.nav.go_to_users()
2016-09-16 14:01:15 - robottelo.ui.browser - DEBUG - mouseMoveTo: {'element': '0.
  ↳8036987570003233-1'}

In [3]: session.ui.user.click(locators.locators.users.new)
2016-09-16 14:01:46 - robottelo.ui.browser - DEBUG - clickElement: {'id': '0.
  ↳12969267888817115-2'}

In [4]: session.ui.user.assign_value(locators.locators.users.username, "my_username")
2016-09-16 14:02:13 - robottelo.ui.browser - DEBUG - sendKeysToElement: {'id': '0.
  ↳12969267888817115-3', 'value': 'my_username'}

In [5]: exit
2016-09-16 14:05:46 - robottelo.ui.browser - DEBUG - logout
2016-09-16 14:05:46 - robottelo.ui.browser - DEBUG - Close Browser
```

While you interact with the UI using the helpers as the ones in the example above you see your browser window changing interactively, if you prefer to use a docker browser it is possible to connect via VNC or get screenshots calling `session.browser.save_screenshot()`

It is also possible to open the *browse* session in specific page if you specify the entity name

```
# opens the session with browser already in users page
(robottelo_env)[you@host robottelo]$ manage ui browse user

# create user using factory
In [1]: session.ui.make_user(username="my_username")
```

3.4.2 Decorators

This section explains Robottelo decorators.

Contents

- *Decorators*
 - *Modules*
 - *stubbed*
 - *skip_if_os*
 - *tier[n]*
 - *skip_if_not_set*
 - *cacheable*
 - *run_in_one_thread*

Modules

Robottelo decorators are located under *decorator package*. Most of them are used to control if a test must be skipped or executed accordingly with specific configurations.

stubbed

stubbed skips any test it decorates. A reason can be provided as parameter and its default is “Not Implemented”. Most of times it is used to define all manual (not automated) tests related to a feature. Example:

```
from robottelo.decorators import stubbed

@stubbed()
def test_negative_create_matcher_attribute_priority(self):
    """Test will be implemented later"""

@stubbed('Some other reason than not implemented')
def test_positive_create_matcher_attribute_priority(self):
    """Test will be implemented later"""
```

Please note that “stubbed” is a decorator generator, and cannot be used as a “classic” Python decorator - it must be ‘@stubbed()’, not ‘@stubbed’ (note the parenthesis).

skip_if_os

skip_if_os skips test based on Foreman/Satellite host os version. It communicates with host defined on `robottelo.properties` to get its os version. Currently it checks only Red Hat Enterprise Linux versions. Example:

```
from robottelo.decorators.host import skip_if_os

@skip_if_os('RHEL6')
def test_positive_create_custom_ostree_repo(self):
    """Create Custom ostree repository"""

@skip_if_os('RHEL6', 'RHEL5')
def test_negative_create_custom_ostree_repo(self):
    """Create Custom ostree repository"""
```

The first test will be skipped if host os is RHEL6.x.y, where x and y can be any number. If RHEL6.1 was used as parameter it would skip for any RHEL6.1.z version and so on.

Arbitrary number versions can be passed as parameters. On second test both RHEL 5 and 6 families would be skipped.

This decorator is used to avoid false failures when an feature is supported only on one os version. For example, ostree repository is available in RHEL7 but not in RHEL6.

tier[n]

This set of decorators defines test levels:

- tier1 marks component level functional tests (that verify defined functional requirements using a range of normal and erroneous input data). Example:

```
from robottelo.decorators import tier1

@tier1
def test_positive_create_with_username(self):
    """Create User for all variations of Username"""
```

- tier2 marks integration level functional tests and may include basic non-functional tests (security, performance regression, installation, compose validation). Example:

```
from robottelo.decorators import tier2

@tier2
def test_positive_view_cve(self):
    """View CVE number(s) in Errata Details page"""
```

- tier3 marks system level tests:

```
from robottelo.decorators import tier3

@tier3
def test_positive_sync_with_enabled_notification(self):
    """Receive email after every sync operation"""
```

- tier4 marks complex and long running tests. Example:

```
from robottelo.decorators import tier4

@tier4
def test_positive_upload_to_satellite(self):
    """Perform end to end oscap test and upload reports"""
```

skip_if_not_set

skip_if_not_set skips test if one or more specified configuration options is not set in robottelo.properties. It is used to define tests specific to a selected (optional) feature. Without the decorator, such tests would fail if the tested feature is not enabled. Example:

```
from robottelo.decorators import skip_if_not_set

@skip_if_not_set('ldap')
```

(continues on next page)

(continued from previous page)

```
def test_positive_ldap_auth_usergroup_user_add(self):  
    """New user added to UserGroup inherits roles"""
```

cacheable

cacheable makes an optional object cache available. This is used when creating factory objects for CLI tests. For example:

```
from robottelo.decorators import cacheable  
  
@cacheable  
def make_role(options=None):  
    """create a role using ``hammer role create``"""
```

run_in_one_thread

run_in_one_thread defines test that cannot be run in parallel with other tests. This is useful for preventing conflicts between tests that interact with the same component. Example:

```
from robottelo.decorators import run_in_one_thread  
  
@run_in_one_thread  
def test_positive_delete_manifest(self):  
    """Check if deleting a manifest removes it from Activation key"""
```

3.4.3 SSH (Secure SHell)

This section explains Robottelo's ssh module.

Contents

- *SSH (Secure SHell)*
 - *Introduction*
 - *SSHCommandResult*
 - *Main Functions*
 - *Helper Functions*

Introduction

Robottelo uses ssh extensively to access remote servers. Functions from `robottelo.ssh` make ssh access easier and are explained on next sections.

SSHCommandResult

SSHCommandResult represents the result of a ssh command. It holds `stdout` on attribute with same name. `stderr` and `return_code` are available the same way. An example of typical result is presented below:


```
SSHCommandResult (
    stdout=['Red Hat Enterprise Linux Server release 7.2 (Maipo)', ''],
    stderr='',
    return_code=0,
    output_format=None
)
```

Attribute `output_format` can be `None`, `csv` or `json`. The former two options are heavily used to define output format on Hammer CLI.

Main Functions

`command` is the main function of `ssh` module. Its full signature is:

```
def command(cmd, hostname=None, output_format=None, username=None,
            password=None, key_filename=None, timeout=10):
    """Executes SSH command(s) on remote hostname."""
```

Most of parameters can be read from `robottelo.properties` file. So in a properly configured project commands can be easily executed on host. The function returns a `SSHCommandResult` instance. The following code was executed to generate the example of first section of this document:

```
>>> from robottelo import ssh
>>> print(ssh.command('cat /etc/redhat-release'))
SSHCommandResult (
    stdout=['Red Hat Enterprise Linux Server release 7.2 (Maipo)', ''],
    stderr='',
    return_code=0,
    output_format=None
)
```

Another important function is `execute_command`. It is similar to `command`. But it reuses an existing connection. So several commands can be executed on a single connection, saving resources and execution time:

```
with ssh.get_connection(..) as connection:
    ssh.execute_command('cp /origin /destiny', connection)
    ssh.execute_command('chmod 0777 /destiny', connection)
    ssh.execute_command("echo 'foo' > /destiny/bar")
```

Helper Functions

The module provide some helper functions which use `command` internally for common ssh operations:

- **add_authorized_key**: Add public key to remote authorized keys;
- **upload_file**: Upload file to remote host;
- **download_file**: Download file from remote host;
- **is_ssh_pub_key**: Validate public key.

3.5 API Reference

This page contains auto-generated API reference documentation¹.

3.5.1 robottelo

This module contains helper code used by `tests.foreman` module.

This module is subservient to `tests.foreman`, and exists solely for the sake of helping that module get its work done. For example, `tests.foreman.cli` relies upon `robottelo.cli`. More generally: code in `tests` calls code in `robottelo`, but not the other way around.

Subpackages

`robottelo.api`

Submodules

`robottelo.api.assertions`

Module Contents

Functions

<code>assert_api_not_raises(expected_exception, callable_obj=None, expected_value=None, value_handler=None, *args, **kwargs)</code>	Fail if an exception of class <code>expected_exception</code> is raised by
---	--

<code>assert_api_not_raises_regex(expected_exception, expected_regex, callable_obj=None, expected_value=None, value_handler=None, *args, **kwargs)</code>	Fail if an exception of class <code>expected_exception</code> is raised and the
---	---

`robottelo.api.assertions.assert_api_not_raises` (*expected_exception*, *callable_obj=None*, *expected_value=None*, *value_handler=None*, **args*, ***kwargs*)

Fail if an exception of class `expected_exception` is raised by `callableObj` when invoked with specified positional and keyword arguments. If a different type of exception is raised, it will not be caught, and the test case will be deemed to have suffered an error, exactly as for an unexpected exception.

If called with `callableObj` omitted or `None`, will return a context object used like this:

```
with assert_api_not_raises(SomeException):
    do_something()
```

The context manager keeps a reference to the exception as the 'exception' attribute. This allows you to inspect the exception after the assertion:

¹ Created with `sphinx-autoapi`

```

with assert_api_not_raises(SomeException) as cm:
    do_something()
the_exception = cm.exception
assert the_exception.error_code == 1

```

In addition, optional 'http_status_code' arg may be passed. This allows to specify exact HTTP status code, returned by `requests.HTTPError`, which should be validated. In such case only expected exception with expected response code will be caught.

```

robottelo.api.assertions.assert_api_not_raises_regex(expected_exception,
                                                    expected_regex,
                                                    callable_obj=None,      ex-
                                                    expected_value=None,
                                                    value_handler=None,    *args,
                                                    **kwargs)

```

Fail if an exception of class `expected_exception` is raised and the message in the exception matches a regex.

robottelo.api.utils

Module containing convenience functions for working with the API.

Module Contents

Classes

<code>templateupdate(temp)</code>	Context Manager to unlock lock template for updating
-----------------------------------	--

Functions

<code>call_entity_method_with_timeout(entity_callable, timeout=300, **kwargs)</code>	Call Entity callable with a custom timeout
<code>enable_rhrepo_and_fetchid(basearch, org_id, product, repo, reposet, releasever)</code>	Enable a RedHat Repository and fetches it's Id.
<code>promote(content_view_version, environment_id, force=False)</code>	Call <code>content_view_version.promote(...)</code> .
<code>upload_manifest(organization_id, manifest)</code>	Call <code>nailgun.entities.Subscription.upload</code> .
<code>publish_puppet_module(puppet_modules, repo_url, organization_id=None)</code>	Creates puppet repo, sync it via provided url and publish using
<code>delete_puppet_class(puppetclass_name, puppet_module=None, proxy_hostname=None, environment_name=None)</code>	Removes puppet class entity and uninstall puppet module from Capsule if
<code>create_sync_custom_repo(org_id=None, product_name=None, repo_name=None, repo_url=None, repo_type=None, repo_unprotected=True, docker_upstream_name=None)</code>	Create product/repo, sync it and returns repo_id
<code>enable_sync_redhat_repo(rh_repo, org_id, timeout=1500)</code>	Enable the RedHat repo, sync it and returns repo_id

Continued on next page

Table 3 – continued from previous page

<code>cv_publish_promote(name=None, env_name=None, repo_id=None, org_id=None)</code>	Create, publish and promote CV to selected environment
<code>one_to_one_names(name)</code>	Generate the names Satellite might use for a one to one field.
<code>one_to_many_names(name)</code>	Generate the names Satellite might use for a one to many field.
<code>configure_provisioning(org=None, loc=None, compute=False, os=None)</code>	Create and configure org, loc, product, repo, cv, env. Update proxy,
<code>create_role_permissions(role, permissions_types_names, search=None)</code>	Create role permissions found in dict permissions_types_names.
<code>wait_for_tasks(search_query, search_rate=1, max_tries=10, poll_rate=None, poll_timeout=None)</code>	Search for tasks by specified search query and poll them to ensure that
<code>wait_for_syncplan_tasks(repo_backend_id=None, timeout=10, repo_name=None)</code>	Search the pulp tasks and identify repositories sync tasks with
<code>wait_for_errata_applicability_task(host_id, from_when, search_rate=1, max_tries=10, poll_rate=None, poll_timeout=15)</code>	Search the generate applicability task for given host and make sure it finishes
<code>create_discovered_host(name=None, ip_address=None, mac_address=None, options=None)</code>	Creates a discovered host.
<code>update_vm_host_location(vm_client, location_id)</code>	Update vm client host location.
<code>check_create_os_with_title(os_title)</code>	Check if the OS is present, if not create the required OS
<code>attach_custom_product_subscription(prod_name=None, host_name=None)</code>	Attach custom product subscription to client host
<code>update_provisioning_template(name=None, old=None, new=None)</code>	Update provisioning template content
<code>apply_package_filter(content_view, repo, package, inclusion=True)</code>	Apply package filter on content view
<code>create_org_admin_role(orgs, locs, name=None)</code>	Helper function to create org admin role for particular
<code>create_org_admin_user(orgs, locs)</code>	Helper function to create an Org Admin user by assigning org admin role and assign
<code>skip_yum_update_during_provisioning(template=None, reverse=False)</code>	Hide the yum update command with echo text
<code>set_hammer_api_timeout(timeout=-1, reverse=False)</code>	Set hammer API request timeout on Satellite

`robottelo.api.utils.call_entity_method_with_timeout(entity_callable, timeout=300, **kwargs)`

Call Entity callable with a custom timeout

:param entity_callable, the entity method object to call :param timeout: the time to wait for the method call to finish :param kwargs: the kwargs to pass to the entity callable

Usage:

`call_entity_method_with_timeout(entities.Repository(id=repo_id).sync, timeout=1500)`

`robottelo.api.utils.enable_rhrepo_and_fetchid(basearch, org_id, product, repo, reposit, releasever)`

Enable a RedHat Repository and fetches it's Id.

Parameters

- **org_id** (*str*) – The organization Id.
- **product** (*str*) – The product name in which repository exists.
- **reposit** (*str*) – The reposit name in which repository exists.

- **repo** (*str*) – The repository name whose Id is to be fetched.
- **basearch** (*str*) – The architecture of the repository.
- **optional releasever** (*str*) – The releasever of the repository.

Returns Returns the repository Id.

Return type *str*

`robottelo.api.utils.promote(content_view_version, environment_id, force=False)`
Call `content_view_version.promote(...)`.

Parameters

- **content_view_version** – A `nailgun.entities.ContentViewVersion` object.
- **environment_id** – An environment ID.
- **force** – Whether to force the promotion or not. Only needed if promoting to a lifecycle environment that is not the next in order of sequence.

Returns Whatever `nailgun.entities.ContentViewVersion.promote` returns.

`robottelo.api.utils.upload_manifest(organization_id, manifest)`
Call `nailgun.entities.Subscription.upload`.

Parameters

- **organization_id** – An organization ID.
- **manifest** – A file object referencing a Red Hat Satellite 6 manifest.

Returns Whatever `nailgun.entities.Subscription.upload` returns.

`robottelo.api.utils.publish_puppet_module(puppet_modules, repo_url, organization_id=None)`

Creates puppet repo, sync it via provided url and publish using Content View publishing mechanism. It makes puppet class available via Puppet Environment created by Content View and returns Content View entity.

Parameters

- **puppet_modules** – List of dictionaries with module ‘author’ and module ‘name’ fields.
- **repo_url** (*str*) – Url of the repo that can be synced using pulp: pulp repo or puppet forge.
- **organization_id** – Organization id that is shared between created entities.

Returns `nailgun.entities.ContentView` entity.

`robottelo.api.utils.delete_puppet_class(puppetclass_name, puppet_module=None, proxy_hostname=None, environment_name=None)`

Removes puppet class entity and uninstall puppet module from Capsule if puppet module name and Capsule details provided.

Parameters

- **puppetclass_name** (*str*) – Name of the puppet class entity that should be removed.
- **puppet_module** (*str*) – Name of the module that should be uninstalled via puppet.
- **proxy_hostname** (*str*) – Hostname of the Capsule from which puppet module should be removed.
- **environment_name** (*str*) – Name of environment where puppet module was imported.

```
robottelo.api.utils.create_sync_custom_repo (org_id=None, product_name=None,
                                             repo_name=None, repo_url=None,
                                             repo_type=None, repo_unprotected=True,
                                             docker_upstream_name=None)
```

Create product/repo, sync it and returns repo_id

```
robottelo.api.utils.enable_sync_redhat_repo (rh_repo, org_id, timeout=1500)
```

Enable the RedHat repo, sync it and returns repo_id

```
robottelo.api.utils.cv_publish_promote (name=None, env_name=None, repo_id=None,
                                         org_id=None)
```

Create, publish and promote CV to selected environment

```
robottelo.api.utils.one_to_one_names (name)
```

Generate the names Satellite might use for a one to one field.

Example of usage:

```
>>> one_to_many_names('person') == {'person_name', 'person_id'}
True
```

Parameters *name* – A field name.

Returns A set including both name and variations on name.

```
robottelo.api.utils.one_to_many_names (name)
```

Generate the names Satellite might use for a one to many field.

Example of usage:

```
>>> one_to_many_names('person') == {'person', 'person_ids', 'people'}
True
```

Parameters *name* – A field name.

Returns A set including both name and variations on name.

```
robottelo.api.utils.configure_provisioning (org=None, loc=None, compute=False,
                                             os=None)
```

Create and configure org, loc, product, repo, cv, env. Update proxy, domain, subnet, compute resource, provision templates and medium with previously created entities and create a hostgroup using all mentioned entities.

Parameters

- **org** (*str*) – Default Organization that should be used in both host discovering and host provisioning procedures
- **loc** (*str*) – Default Location that should be used in both host discovering and host provisioning procedures
- **compute** (*bool*) – If False creates a default Libvirt compute resource
- **os** (*str*) – Specify the os to be used while provisioning and to associate related entities to the specified os.

Returns List of created entities that can be re-used further in provisioning or validation procedure (e.g. hostgroup or domain)

```
robottelo.api.utils.create_role_permissions (role, permissions_types_names,
                                             search=None)
```

Create role permissions found in dict permissions_types_names.

Parameters

- **role** – `nailgun.entities.Role`
- **permissions_types_names** – a dict containing resource types and permission names to add to the role.
- **search** – string that contains search criteria that should be applied to the filter

example usage:

```
permissions_types_names = {
    '(Miscellaneous)': ['access_dashboard'],
    'Organization': ['view_organizations'],
    'Location': ['view_locations'],
    'Katello::KTEEnvironment': [
        'view_lifecycle_environments',
        'edit_lifecycle_environments',
        'promote_or_remove_content_views_to_environments'
    ]
}
role = entities.Role(name='example_role_name').create()
create_role_permissions(
    role,
    permissions_types_names,
    'name = {0}'.format(lce.name)
)
```

`robottelo.api.utils.wait_for_tasks` (*search_query*, *search_rate=1*, *max_tries=10*,
poll_rate=None, *poll_timeout=None*)
Search for tasks by specified search query and poll them to ensure that task has finished.

Parameters

- **search_query** – Search query that will be passed to API call.
- **search_rate** – Delay between searches.
- **max_tries** – How many times search should be executed.
- **poll_rate** – Delay between the end of one task check-up and the start of the next check-up. Parameter for `nailgun.entities.ForemanTask.poll()` method.
- **poll_timeout** – Maximum number of seconds to wait until timing out. Parameter for `nailgun.entities.ForemanTask.poll()` method.

Returns List of `nailgun.entities.ForemanTasks` entities.

Raises `AssertionError`. If not tasks were found until timeout.

`robottelo.api.utils.wait_for_syncplan_tasks` (*repo_backend_id=None*, *timeout=10*,
repo_name=None)

Search the pulp tasks and identify repositories sync tasks with specified name or backend_identifier

Parameters

- **repo_backend_id** – The Backend ID for the repository to identify the repo in Pulp environment
- **timeout** – Value to decided how long to check for the Sync task
- **repo_name** – If `repo_backend_id` can not be passed, pass the `repo_name`

```
robottelo.api.utils.wait_for_errata_applicability_task (host_id, from_when,
                                                       search_rate=1,
                                                       max_tries=10,
                                                       poll_rate=None,
                                                       poll_timeout=15)
```

Search the generate applicability task for given host and make sure it finishes

Parameters

- **host_id** (*int*) – Content host ID of the host where we are regenerating applicability.
- **from_when** (*int*) – Timestamp (in UTC) to limit number of returned tasks to investigate.
- **search_rate** (*int*) – Delay between searches.
- **max_tries** (*int*) – How many times search should be executed.
- **poll_rate** (*int*) – Delay between the end of one task check-up and the start of the next check-up. Parameter for `nailgun.entities.ForemanTask.poll()` method.
- **poll_timeout** (*int*) – Maximum number of seconds to wait until timing out. Parameter for `nailgun.entities.ForemanTask.poll()` method.

Returns Relevant errata applicability task.

Raises `AssertionError`. If not tasks were found for given host until timeout.

```
robottelo.api.utils.create_discovered_host (name=None, ip_address=None,
                                           mac_address=None, options=None)
```

Creates a discovered host.

Parameters

- **name** (*str*) – Name of discovered host.
- **ip_address** (*str*) – A valid ip address.
- **mac_address** (*str*) – A valid mac address.
- **options** (*dict*) – additional facts to add to discovered host

Returns dict of `entities.DiscoveredHost` facts.

```
robottelo.api.utils.update_vm_host_location (vm_client, location_id)
```

Update vm client host location.

Parameters

- **vm_client** – A subscribed Virtual Machine client instance.
- **location_id** – The location id to update the `vm_client` host with.

```
robottelo.api.utils.check_create_os_with_title (os_title)
```

Check if the OS is present, if not create the required OS

Parameters `os_title` – OS title to check, and create (like: RedHat 7.5)

Returns Created or found OS

```
robottelo.api.utils.attach_custom_product_subscription (prod_name=None,
                                                       host_name=None)
```

Attach custom product subscription to client host :param str prod_name: custom product name :param str host_name: client host name

```
class robottelo.api.utils.templateupdate (temp)
```

Context Manager to unlock lock template for updating

`__enter__` (*self*)
 Unlocks template for update

`__exit__` (*self, exc_type, exc_val, exc_tb*)
 Locks template after update

`robottelo.api.utils.update_provisioning_template` (*name=None, old=None, new=None*)
 Update provisioning template content

Parameters

- **name** (*str*) – template provisioning name
- **old** (*str*) – current content
- **new** (*str*) – replace content

Return bool True/False

`robottelo.api.utils.apply_package_filter` (*content_view, repo, package, inclusion=True*)
 Apply package filter on content view

Parameters

- **content_view** – entity content view
- **repo** – entity repository
- **package** (*str*) – package name to filter
- **inclusion** (*bool*) – True/False based on include or exclude filter

:return list : list of content view versions

`robottelo.api.utils.create_org_admin_role` (*orgs, locs, name=None*)
 Helper function to create org admin role for particular organizations and locations by cloning 'Organization admin' role.

Parameters

- **orgs** (*list*) – The list of organizations for which the org admin is being created
- **locs** (*list*) – The list of locations for which the org admin is being created
- **name** (*str*) – The name of cloned Org Admin role, autogenerates if None provided

Return dict The object of ``nailgun.Role`` of Org Admin role.

`robottelo.api.utils.create_org_admin_user` (*orgs, locs*)
 Helper function to create an Org Admin user by assigning org admin role and assign taxonomies to Role and User

The taxonomies for role and user will be assigned based on parameters of this function

Return User Returns the ``nailgun.entities.User`` object with passwd attr

`robottelo.api.utils.skip_yum_update_during_provisioning` (*template=None, reverse=False*) *re-*

Hides the yum update command with echo text

Parameters

- **template** (*str*) – The template name where the yum update will be hidden
- **reverse** (*bool*) – Reverses the echo text to yum update

Returns Boolean True on success else exception

`robottelo.api.utils.set_hammer_api_timeout (timeout=-1, reverse=False)`
Set hammer API request timeout on Satellite

Parameters

- **timeout** (*int*) – request timeout in seconds
- **reverse** (*bool*) – Reverses the request timeout

Returns `ssh.command`

robottelo.cli

Submodules

robottelo.cli.activationkey

Usage:

```
hammer activation-key [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

<code>add-host-collection</code>	Associate a resource
<code>add-subscription</code>	Add subscription
<code>content-override</code>	Override product content defaults
<code>copy</code>	Copy an activation key
<code>create</code>	Create an activation key
<code>delete</code>	Destroy an activation key
<code>host-collections</code>	List associated host collections
<code>info</code>	Show an activation key
<code>list</code>	List activation keys
<code>product-content</code>	List associated products
<code>remove-host-collection</code>	Disassociate a resource
<code>remove-subscription</code>	Remove subscription
<code>subscriptions</code>	List associated subscriptions
<code>update</code>	Update an activation key

Module Contents

Classes

<code>ActivationKey()</code>	Manipulates Katello's activation-key.
------------------------------	---------------------------------------

class `robottelo.cli.activationkey.ActivationKey`

Bases: `robottelo.cli.base.Base`

Manipulates Katello's activation-key.

```

command_base = activation-key
classmethod add_host_collection (cls, options=None)
    Associate a resource
classmethod add_subscription (cls, options=None)
    Add subscription
classmethod content_override (cls, options=None)
    Override product content defaults
classmethod copy (cls, options=None)
    Copy an activation key
classmethod host_collection (cls, options=None)
    List associated host collections
classmethod product_content (cls, options=None)
    List associated products
classmethod remove_host_collection (cls, options=None)
    Remove the associated resource
classmethod remove_repository (cls, options=None)
    Disassociate a resource
classmethod remove_subscription (cls, options=None)
    Remove subscription
classmethod subscriptions (cls, options=None, output_format=None)
    List associated subscriptions

```

robottelo.cli.ansible

Usage:: ansible [OPTIONS] SUBCOMMAND [ARG] ...

Parameters:: SUBCOMMAND Subcommand [ARG] ... Subcommand arguments

Subcommands:: roles Manage ansible roles variables Manage ansible variables

Module Contents

Classes

Ansible()

Manipulates Ansible Variables and roles.

class robottelo.cli.ansible.**Ansible**

Bases: *robottelo.cli.base.Base*

Manipulates Ansible Variables and roles.

command_base = ansible

classmethod **roles_import** (*cls, options=None*)

Import ansible roles

classmethod **variables_import** (*cls, options=None*)

Import ansible variables

```
classmethod roles_list(cls, options=None)  
    List ansible roles
```

robottelo.cli.architecture

Usage:

```
hammer architecture [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

add_operatingsystem	Associate a resource
create	Create an architecture.
delete	Delete an architecture.
info	Show an architecture.
list	List all architectures.
remove_operatingsystem	Disassociate a resource
update	Update an architecture.

Module Contents

Classes

<i>Architecture()</i>	Manipulates Foreman's architecture.
-----------------------	-------------------------------------

```
class robottelo.cli.architecture.Architecture
```

```
    Bases: robottelo.cli.base.Base
```

```
    Manipulates Foreman's architecture.
```

```
    command_base = architecture
```

robottelo.cli.arfreport

Usage:

```
arf-report [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:: delete Delete an ARF Report download Download bziped ARF report download-html Download ARF report in HTML info Show an ARF report list List ARF reports

Module Contents

Classes

Arfreport()

Manipulates Satellite's arf-report.

class robottelo.cli.arfreport.**Arfreport**

Bases: *robottelo.cli.base.Base*

Manipulates Satellite's arf-report.

command_base = arf-report

classmethod list (*cls, options=None*)

Search arf host reports

Usage:

```
hammer arf-report list [OPTIONS]
```

Options:

```
--location LOCATION_NAME           Location name
--location-id LOCATION_ID           Location ID
--location-title LOCATION_TITLE     Location title
--order ORDER                        Sort field and order, eg. 'id DESC'
--organization ORGANIZATION_NAME    Organization name
--organization-id ORGANIZATION_ID    Organization ID
--organization-title ORGANIZATION_TITLE Organization title
--page PAGE                           Paginate results
--per-page PER_PAGE                  Number of entries per request
--search SEARCH                       Filter results
-h, --help                            Print help
```

robottelo.cli.assertions

Module Contents

Functions

assert_cli_not_raises(expected_exception, callable_obj=None, expected_value=None, value_handler=None, *args, **kwargs) Fail if an exception of class expected_exception is raised by

assert_cli_not_raises_regex(expected_exception, expected_regex, callable_obj=None, expected_value=None, value_handler=None, *args, **kwargs) Fail if an exception of class expected_exception is raised and the

```
robottelo.cli.assertions.assert_cli_not_raises (expected_exception,
                                                callable_obj=None,          ex-
                                                pected_value=None,
                                                value_handler=None, *args, **kwargs)
```

Fail if an exception of class `expected_exception` is raised by `callableObj` when invoked with specified positional and keyword arguments. If a different type of exception is raised, it will not be caught, and the test case will be deemed to have suffered an error, exactly as for an unexpected exception.

If called with `callableObj` omitted or `None`, will return a context object used like this:

```
with assert_cli_raises(SomeException):
    do_something()
```

The context manager keeps a reference to the exception as the 'exception' attribute. This allows you to inspect the exception after the assertion:

```
with assert_cli_raises(SomeException) as cm:
    do_something()
the_exception = cm.exception
self.assertEqual(the_exception.error_code, 1)
```

In addition, optional 'cli_return_code' arg may be passed. This allows to specify exact CLI return code `robottelo.cli.base.CLIReturnCodeError` accordingly, which should be validated. In such case only expected exception with expected response code will be caught.

```
robottelo.cli.assertions.assert_cli_not_raises_regex (expected_exception,
                                                       expected_regex,
                                                       callable_obj=None,          ex-
                                                       pected_value=None,
                                                       value_handler=None, *args,
                                                       **kwargs)
```

Fail if an exception of class `expected_exception` is raised and the message in the exception matches a regex.

robottelo.cli.auth

Usage:: hammer auth [OPTIONS] SUBCOMMAND [ARG] ...

Parameters:: SUBCOMMAND subcommand [ARG] ... subcommand arguments

Subcommands:: login Set credentials logout Wipe your credentials status Information about current connections

Module Contents

Classes

<code>Auth()</code>	Authenticates Foreman users
<code>AuthLogin()</code>	Auth Login for Foreman CLI

```
class robottelo.cli.auth.Auth
    Bases: robottelo.cli.base.Base
    Authenticates Foreman users
    command_base = auth
```

classmethod login (*cls, options=None*)

Set credentials

classmethod logout (*cls, options=None*)

Wipe credentials

classmethod status (*cls, options=None*)

Show login status

class `robottelo.cli.auth.AuthLogin`

Bases: `robottelo.cli.base.Base`

Auth Login for Foreman CLI

command_base = **auth login**

classmethod basic (*cls, options=None*)

Provide username and password

classmethod oauth (*cls, options=None*)

Supports for both with/without 2fa

`robottelo.cli.base`

Generic base class for cli hammer commands.

Module Contents

Classes

`Base()`

@param `command_base`: base command of hammer.

exception `robottelo.cli.base.CLIError`

Bases: `Exception`

Indicates that a CLI command could not be run.

exception `robottelo.cli.base.CLIBaseError` (*return_code, stderr, msg*)

Bases: `Exception`

Indicates that a CLI command has finished with return code different from zero.

Parameters

- **return_code** – CLI command return code
- **stderr** – contents of the `stderr`
- **msg** – explanation of the error

`__str__` (*self*)

Include class name, `return_code`, `stderr` and `msg` to string repr so `assertRaisesRegexp` can be used to assert error present on any attribute

`__repr__` (*self*)

Include class name `return_code`, `stderr` and `msg` to improve logging

exception `robottelo.cli.base.CLIReturnCodeError` (*return_code, stderr, msg*)

Bases: `robottelo.cli.base.CLIBaseError`

Error to be raised when an error occurs due to some validation error when execution hammer cli. See: <https://github.com/SatelliteQE/robottelo/issues/3790> for more details

exception robottelo.cli.base.**CLIDataBaseError** (*return_code, stderr, msg*)

Bases: *robottelo.cli.base.CLIBaseError*

Error to be raised when an error occurs due to some missing parameter which cause a data base error on hammer See: <https://github.com/SatelliteQE/robottelo/issues/3790> for more details

class robottelo.cli.base.**Base**

Bases: object

@param *command_base*: base command of hammer. Output of recent *hammer -help*:

activation-key	Manipulate activation keys.
architecture	Manipulate architectures.
auth	Foreman connection login/logout.
auth-source	Manipulate auth sources.
bootdisk	Download boot disks
capsule	Manipulate capsule
compute-resource	Manipulate compute resources.
content-host	Manipulate content hosts on the server
content-report	View Content Reports
content-view	Manipulate content views.
defaults	Defaults management
docker	Manipulate docker content
domain	Manipulate domains.
environment	Manipulate environments.
erratum	Manipulate errata
fact	Search facts.
filter	Manage permission filters.
global -parameter	Manipulate global parameters.
gpg	Manipulate GPG Key actions on the server
host	Manipulate hosts.
host-collection	Manipulate host collections
hostgroup	Manipulate hostgroups.
import	Import data exported from a Red Hat Sat..
lifecycle-environment	Manipulate lifecycle_environments
location	Manipulate locations.
medium	Manipulate installation media.
model	Manipulate hardware models.
organization	Manipulate organizations
os	Manipulate operating system.
ostree-branch	Manipulate ostree branches
package	Manipulate packages.
package-group	Manipulate package groups
partition-table	Manipulate partition tables.
ping	Get the status of the server
product	Manipulate products.
proxy	Manipulate smart proxies.
puppet- class	Search puppet modules.
puppet-module	View Puppet Module details.
report	Browse and read reports.
repository	Manipulate repositories
repository-set	Manipulate repository sets on the server
role	Manage user roles.
sc-param	Manipulate smart class parameters .
settings	Change server settings.
shell	Interactive shell

(continues on next page)

(continued from previous page)

subnet	Manipulate subnets.
subscription	Manipulate subscriptions.
sync-plan	Manipulate sync plans
task	Tasks related actions.
template	Manipulate provisioning templates.
user	Manipulate users.
user-group	Manage user groups.

@since: 27.Nov.2013

command_base

command_sub

command_requires_org = False

logger

_db_error_regex

classmethod _handle_response (*cls, response, ignore_stderr=None*)

Verify return_code of the CLI command.

Check for a non-zero return code or any stderr contents.

Parameters

- **response** – a `SSHCommandResult` object, returned by `robottelo.ssh.command`.
- **ignore_stderr** – indicates whether to throw a warning in logs if `stderr` is not empty.

Returns contents of `stdout`.

Raises `robottelo.cli.base.CLIReturnCodeError` – If return code is different from zero.

classmethod add_operating_system (*cls, options=None*)

Adds OS to record.

classmethod create (*cls, options=None, timeout=None*)

Creates a new record using the arguments passed via dictionary.

classmethod delete (*cls, options=None, timeout=None*)

Deletes existing record.

classmethod delete_parameter (*cls, options=None*)

Deletes parameter from record.

classmethod dump (*cls, options=None*)

Displays the content for existing partition table.

classmethod _get_username_password (*cls, username=None, password=None*)

Lookup for the username and password for cli command in following order:

1. user or password parameters
2. `foreman_admin_username` or `foreman_admin_password` attributes
3. `foreman.admin.username` or `foreman.admin.password` configuration

Returns A tuple with the username and password found

Return type tuple

classmethod execute (*cls, command, user=None, password=None, output_format=None, timeout=None, ignore_stderr=None, return_raw_response=None, connection_timeout=None*)

Executes the cli command on the server via ssh

classmethod exists (*cls, options=None, search=None*)

Search for an entity using the query `search[0]="search[1]"`

Will be used the `list` command with the `--search` option to do the search.

If `options` argument already have a search key, then the `search` argument will not be evaluated. Which allows different search query.

classmethod info (*cls, options=None, output_format=None, return_raw_response=None*)

Reads the entity information.

classmethod list (*cls, options=None, per_page=True, output_format='csv'*)

List information. @param options: ID (sometimes name works as well) to retrieve info.

classmethod puppetclasses (*cls, options=None*)

Lists all puppet classes.

classmethod remove_operating_system (*cls, options=None*)

Removes OS from record.

classmethod sc_params (*cls, options=None*)

Lists all smart class parameters.

classmethod set_parameter (*cls, options=None*)

Creates or updates parameter for a record.

classmethod update (*cls, options=None, return_raw_response=None*)

Updates existing record.

classmethod with_user (*cls, username=None, password=None*)

Context Manager for credentials

classmethod _construct_command (*cls, options=None*)

Build a hammer cli command based on the options passed

robottelo.cli.capsule

Usage:

```
hammer capsule [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

content	Manage the capsule content
create	Create a capsule
delete	Delete a capsule
import-classes	Import puppet classes from puppet Capsule.
info	Show a capsule
list	List all capsules

(continues on next page)

(continued from previous page)

refresh-features	Refresh capsule features
update	Update a capsule

Module Contents

Classes

<code>Capsule()</code>	Manipulates Foreman's capsule.
------------------------	--------------------------------

class `robottelo.cli.capsule.Capsule`

Bases: `robottelo.cli.base.Base`

Manipulates Foreman's capsule.

command_base = `capsule`

classmethod `content_add_lifecycle_environment` (*cls, options*)

Add lifecycle environments to the capsule.

classmethod `content_available_lifecycle_environments` (*cls, options*)

List the lifecycle environments not attached to the capsule.

classmethod `content_info` (*cls, options*)

Get current capsule synchronization status.

classmethod `content_lifecycle_environments` (*cls, options*)

List the lifecycle environments attached to the capsule.

classmethod `content_remove_lifecycle_environment` (*cls, options*)

Remove lifecycle environments from the capsule.

classmethod `content_synchronization_status` (*cls, options*)

Get current capsule synchronization status.

classmethod `content_synchronize` (*cls, options, return_raw_response=None, timeout=3600*)

Synchronize the content to the capsule.

classmethod `import_classes` (*cls, options*)

Import puppet classes from puppet Capsule.

classmethod `refresh_features` (*cls, options*)

Refresh capsule features.

`robottelo.cli.computeprofile`

Usage: `hammer compute-profile [OPTIONS] SUBCOMMAND [ARG] ...`

Parameters: `SUBCOMMAND` Subcommand `[ARG] ...` Subcommand arguments

Subcommands: `create` Create a compute profile `delete` Delete a compute profile `info` Show a compute profile `list` List of compute profiles `update` Update a compute profile `values` Create update and delete Compute profile values

Options:

-h, --help Print help Update a compute resource.

Module Contents

Classes

<code>ComputeProfile()</code>	Manipulates Foreman's compute-profile.
-------------------------------	--

class `robottelo.cli.computeprofile.ComputeProfile`

Bases: `robottelo.cli.base.Base`

Manipulates Foreman's compute-profile.

command_base = `compute-profile`

classmethod `values_create` (*cls*, *options=None*)

Create Compute profile values

`robottelo.cli.computeresource`

Usage:

```
hammer compute-resource [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

<code>create</code>	Create a compute resource.
<code>delete</code>	Delete a compute resource.
<code>image</code>	View and manage compute resource's images
<code>info</code>	Show an compute resource.
<code>list</code>	List all compute resources.
<code>update</code>	Update a compute resource.

Module Contents

Classes

<code>ComputeResource()</code>	Manipulates Foreman's compute resources.
--------------------------------	--

class `robottelo.cli.computeresource.ComputeResource`

Bases: `robottelo.cli.base.Base`

Manipulates Foreman's compute resources.

command_base = `compute-resource`

classmethod `image_create` (*cls*, *options*)

Create an image

classmethod `image_info` (*cls*, *options*)

Show an image

classmethod image_available (*cls, options*)

Show images available for addition

classmethod image_delete (*cls, options*)

delete an image

classmethod image_list (*cls, options*)

Show the list of images

classmethod image_update (*cls, options*)

update an image

classmethod networks (*cls, options*)

List available networks for a compute resource

robottelo.cli.contentview

Usage:

```
hammer content-view [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

add-repository	Associate a resource
add-version	Update a content view
component	View and manage components
copy	Copy a content view
create	Create a content view
delete	Delete a content view
filter	View and manage filters
info	Show a content view
list	List content views
publish	Publish a content view
puppet-module	View and manage puppet modules
remove	Remove versions and/or environments from a content view and reassign systems and keys
remove-from-environment	Remove a content view from an environment
remove-repository	Disassociate a resource
remove-version	Remove a content view version from a composite view
update	Update a content view
version	View and manage content view versions

Options:

-h, --help	print help
------------	------------

Module Contents

Classes

<code>ContentViewFilterRule()</code>	Manipulates content view filter rules.
<code>ContentViewFilter()</code>	Manipulates content view filters.
<code>ContentView()</code>	Manipulates Foreman's content view.

class `robottelo.cli.contentview.ContentViewFilterRule`

Bases: `robottelo.cli.base.Base`

Manipulates content view filter rules.

command_base = **content-view filter rule**

classmethod **create** (*cls, options=None*)

Create a content-view filter rule

class `robottelo.cli.contentview.ContentViewFilter`

Bases: `robottelo.cli.base.Base`

Manipulates content view filters.

command_base = **content-view filter**

rule

class `robottelo.cli.contentview.ContentView`

Bases: `robottelo.cli.base.Base`

Manipulates Foreman's content view.

command_base = **content-view**

filter

classmethod **add_repository** (*cls, options*)

Associate repository to a selected CV.

classmethod **add_version** (*cls, options*)

Associate version to a selected CV.

classmethod **copy** (*cls, options*)

Copy existing content-view to a new one

classmethod **publish** (*cls, options, timeout=1500*)

Publishes a new version of content-view.

classmethod **version_info** (*cls, options, output_format=None*)

Provides version info related to content-view's version.

classmethod **version_incremental_update** (*cls, options*)

Performs incremental update of the content-view's version

classmethod **puppet_module_add** (*cls, options*)

Associate puppet_module to selected CV

classmethod **puppet_module_list** (*cls, options*)

List content view puppet modules

classmethod **puppet_module_remove** (*cls, options*)

Remove a puppet module from the content view

classmethod **version_list** (*cls, options*)

Lists content-view's versions.

classmethod `version_promote` (*cls, options, timeout=600*)
Promotes content-view version to next env.

classmethod `version_export` (*cls, options, timeout=300*)
Exports content-view version in given directory

classmethod `version_import` (*cls, options, timeout=300*)
Imports content-view version from a given directory

classmethod `version_delete` (*cls, options*)
Removes content-view version.

classmethod `remove_from_environment` (*cls, options=None*)
Remove content-view from an environment

classmethod `remove` (*cls, options=None*)
Remove versions and/or environments from a content view and reassign content hosts and keys

classmethod `remove_version` (*cls, options=None*)
Remove a content view version from a composite view

classmethod `remove_repository` (*cls, options*)
Remove repository from content view

classmethod `component_add` (*cls, options=None*)
Add components to the content view

classmethod `component_list` (*cls, options=None*)
List components attached to the content view

`robottelo.cli.defaults`

Usage:

```
hammer defaults [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

<code>add</code>	Add a default parameter to config
<code>delete</code>	Delete a default param
<code>list</code>	List <code>all</code> the default parameters
<code>providers</code>	List <code>all</code> the providers

Module Contents

Classes

`Defaults()`

Manipulates Defaults entity

class `robottelo.cli.defaults.Defaults`

Bases: `robottelo.cli.base.Base`

Manipulates Defaults entity

command_base = defaults

classmethod add (*cls, options=None*)

Add parameter to config Usage:

```
hammer defaults add [OPTIONS]
```

Options:

```
--param-name OPTION_NAME      The name of the default option
                               (e.g. organization_id).
--param-value OPTION_VALUE     The value for the default option
--provider OPTION_PROVIDER     The name of the provider providing
                               the value. For list available
                               providers see `hammer defaults
                               providers`.
```

classmethod delete (*cls, options=None*)

Delete parameter from config Usage:

```
hammer defaults delete [OPTIONS]
```

Options:

```
--param-name OPTION_NAME      The name of the default option
```

robottelo.cli.discoveredhost

Usage:

```
hammer discovery [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

```
SUBCOMMAND          subcommand
[ARG] ...           subcommand arguments
```

Subcommands:

```
auto-provision      Auto provision a host
delete              Delete a discovered host
facts               Show a discovered host
info                Show a discovered host
list                List all discovered hosts
provision           Provision a discovered host
reboot              Reboot a host
refresh-facts       Refresh the facts of a host
```

Module Contents

Classes

*DiscoveredHost()*Manipulates Discovery Hosts

class robottelo.cli.discoveredhost.**DiscoveredHost**Bases: *robottelo.cli.base.Base*

Manipulates Discovery Hosts

command_base = discovery**classmethod provision** (*cls, options=None*)

Manually provision discovered host

classmethod facts (*cls, options=None*)

Get all the facts associated with discovered host

robottelo.cli.discoveryrule

Usage:

```
hammer discovery-rule [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create a discovery rule
delete	Delete a rule
info	Show a discovery rule
list	List all discovery rules
update	Update a rule

Module Contents

Classes

*DiscoveryRule()*Manipulates Discovery Rules

class robottelo.cli.discoveryrule.**DiscoveryRule**Bases: *robottelo.cli.base.Base*

Manipulates Discovery Rules

command_base = discovery-rule**robottelo.cli.docker**

Docker related hammer commands

Module Contents

Classes

<code>DockerManifest()</code>	Manipulates Docker manifests
<code>DockerTag()</code>	Manipulates Docker tags
<code>Docker()</code>	Manipulates Docker manifests and tags

class `robottelo.cli.docker.DockerManifest`

Bases: `robottelo.cli.base.Base`

Manipulates Docker manifests

Usage:

```
hammer docker manifest [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

<code>info</code>	Show a docker manifest
<code>list</code>	List docker_manifests

command_base = docker manifest

classmethod `info` (*cls*, *options=None*)

Gets information about docker manifests

Usage:

```
hammer docker manifest info [OPTIONS]
```

Options:

<code>--id ID</code>	a docker manifest identifier
<code>--name NAME</code>	Name to search by
<code>--repository REPOSITORY_NAME</code>	Repository name to search by
<code>--repository-id REPOSITORY_ID</code>	repository ID

classmethod `list` (*cls*, *options=None*, *per_page=True*)

List docker manifests

Usage:

```
hammer docker manifest list [OPTIONS]
```

Options:

<code>--by BY</code>	Field to sort the results on
<code>--content-view CONTENT_VIEW_NAME</code>	Content view name
<code>--content-view-filter CONTENT_VIEW_FILTER_NAME</code>	Name to search by
<code>--content-view-filter-id CONTENT_VIEW_FILTER_ID</code>	<code>filter</code> identifier

(continues on next page)

(continued from previous page)

```

--content-view-id CONTENT_VIEW_ID          content view
                                             numeric identifier
--content-view-version CONTENT_VIEW_VERSION_VERSION Content view
                                             version number
--content-view-version-id CONTENT_VIEW_VERSION_ID Content view
                                             version identifier
--full-results FULL_RESULTS                Whether or not to
                                             show all results
                                             One of true/false,
                                             yes/no, 1/0.
--ids IDS                                  ids to filter
                                             content by
                                             Comma separated
                                             list of values.
--lifecycle-environment LIFECYCLE_ENVIRONMENT_NAME Name to search by
--lifecycle-environment-id LIFECYCLE_ENVIRONMENT_ID
--order ORDER                              Sort field and
                                             order, eg.
                                             "name DESC"
--organization ORGANIZATION_NAME          Organization name
                                             to search by
--organization-id ORGANIZATION_ID         organization ID
--organization-label ORGANIZATION_LABEL   Organization label
                                             to search by
--page PAGE                                Page number,
                                             starting at 1
--per-page PER_PAGE                        Number of results
                                             per page to return
--product PRODUCT_NAME                     Product name to
                                             search by
--product-id PRODUCT_ID                   product numeric
                                             identifier
--repository REPOSITORY_NAME              Repository name to
                                             search by
--repository-id REPOSITORY_ID             repository ID
--search SEARCH                            Search string

```

class robottelo.cli.docker.DockerTagBases: *robottelo.cli.base.Base*

Manipulates Docker tags

Usage:

```
hammer docker tag [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

info	Show a docker tag
list	List docker_tags

command_base = docker tag

classmethod info (*cls, options=None*)

Gets information about docker tags

Usage:

```
hammer docker tag info [OPTIONS]
```

Options:

```
--id ID                a docker tag identifier
--name NAME            Name to search by
--repository REPOSITORY_NAME  Repository name to search by
--repository-id REPOSITORY_ID repository ID
```

classmethod list (*cls, options=None, per_page=True*)

List docker tags

Usage:

```
hammer docker tag list [OPTIONS]
```

Options:

```
--content-view CONTENT_VIEW_NAME      Content view
                                       name
--content-view-filter CONTENT_VIEW_FILTER_NAME  Name to search
                                       by
--content-view-filter-id CONTENT_VIEW_FILTER_ID  filter
                                       identifier
--content-view-id CONTENT_VIEW_ID          content view
                                       numeric
                                       identifier
--content-view-version CONTENT_VIEW_VERSION_VERSION  Content view
                                       version number
--content-view-version-id CONTENT_VIEW_VERSION_ID  Content view
                                       version
                                       identifier
--environment ENVIRONMENT_NAME         Name to search
                                       by
--environment-id ENVIRONMENT_ID
--organization ORGANIZATION_NAME       Organization
                                       name to search
                                       by
--organization-id ORGANIZATION_ID      organization ID
--organization-label ORGANIZATION_LABEL  Organization
                                       label to search
                                       by
--product PRODUCT_NAME                 Product name to
                                       search by
--product-id PRODUCT_ID                product numeric
                                       identifier
--repository REPOSITORY_NAME           Repository name
                                       to search by
--repository-id REPOSITORY_ID          repository ID
```

class `robottelo.cli.docker.Docker`

Bases: `robottelo.cli.base.Base`

Manipulates Docker manifests and tags

Usage:

```
hammer docker [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

container	Manage docker containers
manifest	Manage docker manifests
registry	Manage docker registries
tag	Manage docker tags

command_base = docker

manifest

tag

robottelo.cli.domain

Usage:

```
hammer domain [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create a domain.
delete	Delete a domain.
delete_parameter	Delete parameter for a domain.
info	Show a domain.
list	List of domains
set_parameter	Create or update parameter for a domain.
update	Update a domain.

Module Contents**Classes**

Domain()

Manipulates Foreman's domains.

class robottelo.cli.domain.**Domain**

Bases: *robottelo.cli.base.Base*

Manipulates Foreman's domains.

command_base = domain

robottelo.cli.environment

Usage:

```
hammer environment [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create an environment
delete	Delete an environment
info	Show an environment
list	List all environments
sc-params	List all smart class parameters
update	Update an environment

Module Contents

Classes

<i>Environment()</i>	Manipulates Foreman's environments.
----------------------	-------------------------------------

class robottelo.cli.environment.**Environment**

Bases: *robottelo.cli.base.Base*

Manipulates Foreman's environments.

command_base = environment

classmethod **sc_params** (*cls, options=None*)

List all smart class parameters.

robottelo.cli.erratum

Usage:

```
hammer erratum [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

info	Show an erratum
list	List errata

Module Contents

Classes

<code>Erratum()</code>	Manipulates Foreman's erratum.
------------------------	--------------------------------

```
class robottelo.cli.erratum.Erratum
    Bases: robottelo.cli.base.Base
    Manipulates Foreman's erratum.
    command_base = erratum
```

`robottelo.cli.fact`

Usage:

```
hammer fact [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

<code>list</code>	List all fact values.
-------------------	-----------------------

Module Contents

Classes

<code>Fact()</code>	Searches Foreman's facts.
---------------------	---------------------------

```
class robottelo.cli.fact.Fact
    Bases: robottelo.cli.base.Base
    Searches Foreman's facts.
    command_base = fact
```

`robottelo.cli.factory`

Factory object creation for all CLI methods

Module Contents

Functions

<code>create_object(cli_object, options, values)</code>	Creates <object> with dictionary of arguments.
<code>_entity_with_credentials(credentials, cli_entity_cls)</code>	Create entity class using credentials. If credentials is None will
<code>make_activation_key(options=None)</code>	Creates an Activation Key
<code>make_architecture(options=None)</code>	Creates an Architecture
<code>make_content_view(options=None)</code>	Creates a Content View
<code>make_content_view_with_credentials(options=None, credentials=None)</code>	Helper function to create CV with credentials
<code>make_content_view_filter(options=None)</code>	Creates a Content View Filter
<code>make_content_view_filter_rule(options=None)</code>	Creates a Content View Filter Rule
<code>make_discoveryrule(options=None)</code>	Creates a Discovery Rule
<code>make_gpg_key(options=None)</code>	Creates a GPG Key
<code>make_location(options=None)</code>	Creates a Location
<code>make_model(options=None)</code>	Creates a Hardware Model
<code>make_partition_table(options=None)</code>	Creates a Partition Table
<code>make_product(options=None)</code>	Creates a Product
<code>make_product_with_credentials(options=None, credentials=None)</code>	Helper function to create product with credentials
<code>make_product_wait(options=None, wait_for=5)</code>	Wrapper function for make_product to make it wait before erroring out.
<code>make_proxy(options=None)</code>	Creates a Proxy
<code>make_repository(options=None)</code>	Creates a Repository
<code>make_repository_with_credentials(options=None, credentials=None)</code>	Helper function to create Repository with credentials
<code>make_role(options=None)</code>	Creates a Role
<code>make_filter(options=None)</code>	Creates a Role Filter
<code>make_scap_policy(options=None)</code>	Creates a Scap Policy
<code>make_subnet(options=None)</code>	Creates a Subnet
<code>make_sync_plan(options=None)</code>	Creates a Sync Plan
<code>make_host(options=None)</code>	Creates a Host
<code>make_fake_host(options=None)</code>	Wrapper function for make_host to pass all required options for creation
<code>make_host_collection(options=None)</code>	Creates a Host Collection
<code>make_job_invocation(options=None)</code>	Creates a Job Invocation
<code>make_job_template(options=None)</code>	Creates a Job Template
<code>make_user(options=None)</code>	Creates a User
<code>make_usergroup(options=None)</code>	Creates a User Group
<code>make_usergroup_external(options=None)</code>	Creates an External User Group
<code>make_ldap_auth_source(options=None)</code>	Creates an LDAP Auth Source
<code>make_compute_resource(options=None)</code>	Creates a Compute Resource
<code>make_org(options=None)</code>	Creates an Organization
<code>make_org_with_credentials(options=None, credentials=None)</code>	Helper function to create organization with credentials
<code>make_realm(options=None)</code>	Creates a REALM
<code>make_report_template(options=None)</code>	Creates a Report Template
<code>make_os(options=None)</code>	Creates an Operating System
<code>make_scapcontent(options=None)</code>	Creates Scap Content
<code>make_domain(options=None)</code>	Creates a Domain

Continued on next page

Table 23 – continued from previous page

<code>make_hostgroup(options=None)</code>	Creates a Hostgroup
<code>make_medium(options=None)</code>	Creates a Medium
<code>make_environment(options=None)</code>	Creates a Puppet Environment
<code>make_lifecycle_environment(options=None)</code>	Creates a Lifecycle Environment
<code>make_tailoringfile(options=None)</code>	Creates a tailoring File
<code>make_template(options=None)</code>	Creates a Template
<code>make_template_input(options=None)</code>	Creates Template Input
<code>make_virt_who_config(options=None)</code>	Creates a Virt Who Configuration
<code>activationkey_add_subscription_to_repo(options=None)</code>	Creates a subscription that adds subscription to an activation key
<code>setup_org_for_a_custom_repo(options=None)</code>	Sets up Org for the given custom repo by:
<code>_setup_org_for_a_rh_repo(options=None)</code>	Sets up Org for the given Red Hat repository by:
<code>setup_org_for_a_rh_repo(options=None, force_manifest_upload=False, force_use_cdn=False)</code>	Wrapper above <code>_setup_org_for_a_rh_repo</code> to use custom downstream repo
<code>configure_env_for_provision(org=None, loc=None)</code>	Create and configure org, loc, product, repo, env. Update proxy,
<code>publish_puppet_module(puppet_modules, repo_url, organization_id=None)</code>	Creates puppet repo, sync it via provided url and publish using
<code>setup_virtual_machine(vm, org_label, rh_repos_id=None, repos_label=None, product_label=None, lce=None, activation_key=None, patch_os_release_distro=None, install_katello_agent=True)</code>	Setup a Virtual machine with basic components and tasks.
<code>_get_capsule_vm_distro_repos(distro)</code>	Return the right RH repos info for the capsule setup
<code>add_role_permissions(role_id, resource_permissions)</code>	Create role permissions found in resource permissions dict
<code>setup_cdn_and_custom_repositories(org_id, repos, download_policy='on_demand', synchronize=True)</code>	Setup cdn and custom repositories
<code>setup_cdn_and_custom_repos_content(org_id, lce_id=None, repos=None, upload_manifest=True, download_policy='on_demand', rh_subscriptions=None, default_cv=False)</code>	Setup cdn and custom repositories, content view and activations key
<code>vm_setup_ssh_config(vm, ssh_key_name, host, user=None)</code>	Create host entry in vm ssh config and know_hosts files to allow vm
<code>vm_upload_ssh_key(vm, source_key_path, destination_key_name)</code>	Copy ssh key to virtual machine ssh path and ensure proper permission is
<code>virt_who_hypervisor_config(config_id, virt_who_vm, org_id=None, lce_id=None, hypervisor_hostname=None, configure_ssh=False, hypervisor_user=None, subscription_name=None, exec_one_shot=False, upload_manifest=True, extra_repos=None)</code>	Configure virtual machine as hypervisor virt-who service
<code>make_http_proxy(options=None)</code>	Creates a HTTP Proxy

`robottelo.cli.factory.logger`

`robottelo.cli.factory.ORG_KEYS = ['organization', 'organization-id', 'organization-label']`

`robottelo.cli.factory.CONTENT_VIEW_KEYS = ['content-view', 'content-view-id']`

`robottelo.cli.factory.LIFECYCLE_KEYS = ['lifecycle-environment', 'lifecycle-environment-id']`

exception `robottelo.cli.factory.CLIFactoryError`

Bases: Exception

Indicates an error occurred while creating an entity using hammer

`robottelo.cli.factory.create_object` (*cli_object, options, values*)

Creates <object> with dictionary of arguments.

Parameters

- **cli_object** – A valid CLI object.
- **options** (*dict*) – The default options accepted by the cli_object create
- **values** (*dict*) – Custom values to override default ones.

Raises `robottelo.cli.factory.CLIFactoryError` – Raise an exception if object cannot be created.

Return type dict

Returns A dictionary representing the newly created resource.

`robottelo.cli.factory._entity_with_credentials` (*credentials, cli_entity_cls*)

Create entity class using credentials. If credentials is None will return cli_entity_cls itself

Parameters

- **credentials** – tuple (login, password)
- **cli_entity_cls** – Cli Entity Class

Returns Cli Entity Class

`robottelo.cli.factory.make_activation_key` (*options=None*)

Creates an Activation Key

Parameters **options** – Check options using *hammer activation-key create -help* on satellite.

:returns ActivationKey object

`robottelo.cli.factory.make_architecture` (*options=None*)

Creates an Architecture

Parameters **options** – Check options using *hammer architecture create -help* on satellite.

:returns Architecture object

`robottelo.cli.factory.make_content_view` (*options=None*)

Creates a Content View

Parameters **options** – Check options using *hammer content-view create -help* on satellite.

:returns ContentView object

`robottelo.cli.factory.make_content_view_with_credentials` (*options=None, credentials=None*)

Helper function to create CV with credentials

If credentials is None, the default credentials in robottelo.properties will be used.

`robottelo.cli.factory.make_content_view_filter` (*options=None*)

Creates a Content View Filter

Parameters **options** – Check options using *hammer content-view filter create -help* on satellite.

:returns ContentViewFilter object

`robottelo.cli.factory.make_content_view_filter_rule` (*options=None*)

Creates a Content View Filter Rule

Parameters **options** – Check options using *hammer content-view filter rule create -help* on satellite.

:returns ContentViewFilterRule object

`robottelo.cli.factory.make_discoveryrule (options=None)`
Creates a Discovery Rule

Parameters **options** – Check options using *hammer discovery-rule create –help* on satellite.

:returns DiscoveryRule object

`robottelo.cli.factory.make_gpg_key (options=None)`
Creates a GPG Key

Parameters **options** – Check options using *hammer gpg create –help* on satellite.

:returns GPGKey object

`robottelo.cli.factory.make_location (options=None)`
Creates a Location

Parameters **options** – Check options using *hammer location create –help* on satellite.

:returns Location object

`robottelo.cli.factory.make_model (options=None)`
Creates a Hardware Model

Parameters **options** – Check options using *hammer model create –help* on satellite.

:returns Model object

`robottelo.cli.factory.make_partition_table (options=None)`
Creates a Partition Table

Parameters **options** – Check options using *hammer partition-table create –help* on satellite.

:returns PartitionTable object

`robottelo.cli.factory.make_product (options=None)`
Creates a Product

Parameters **options** – Check options using *hammer product create –help* on satellite.

:returns Product object

`robottelo.cli.factory.make_product_with_credentials (options=None, credentials=None)`

Helper function to create product with credentials

`robottelo.cli.factory.make_product_wait (options=None, wait_for=5)`
Wrapper function for `make_product` to make it wait before erroring out.

This is a temporary workaround for BZ#1332650: Sometimes cli product create errors for no reason when there are multiple product creation requests at the sametime although the product entities are created. This workaround will attempt to wait for 5 seconds and query the product again to make sure it is actually created. If it is not found, it will fail and stop.

Note: This wrapper method is created instead of patching `make_product` because this issue does not happen for all entities and this workaround should be removed once the root cause is identified/fixd.

`robottelo.cli.factory.make_proxy (options=None)`
Creates a Proxy

Parameters **options** – Check options using *hammer proxy create –help* on satellite.

:returns Proxy object

`robottelo.cli.factory.make_repository (options=None)`

Creates a Repository

Parameters `options` – Check options using *hammer repository create –help* on satellite.

:returns Repository object

`robottelo.cli.factory.make_repository_with_credentials (options=None, credentials=None)`

Helper function to create Repository with credentials

`robottelo.cli.factory.make_role (options=None)`

Creates a Role

Parameters `options` – Check options using *hammer role create –help* on satellite.

:returns Role object

`robottelo.cli.factory.make_filter (options=None)`

Creates a Role Filter

Parameters `options` – Check options using *hammer filter create –help* on satellite.

:returns Role object

`robottelo.cli.factory.make_scap_policy (options=None)`

Creates a Scap Policy

Parameters `options` – Check options using *hammer policy create –help* on satellite.

:returns Scappolicy object

`robottelo.cli.factory.make_subnet (options=None)`

Creates a Subnet

Parameters `options` – Check options using *hammer subnet create –help* on satellite.

:returns Subnet object

`robottelo.cli.factory.make_sync_plan (options=None)`

Creates a Sync Plan

Parameters `options` – Check options using *hammer sync-plan create –help* on satellite.

:returns SyncPlan object

`robottelo.cli.factory.make_host (options=None)`

Creates a Host

Parameters `options` – Check options using *hammer host create –help* on satellite.

:returns Host object

`robottelo.cli.factory.make_fake_host (options=None)`

Wrapper function for `make_host` to pass all required options for creation of a fake host

`robottelo.cli.factory.make_host_collection (options=None)`

Creates a Host Collection

Parameters `options` – Check options using *hammer host-collection create –help* on satellite.

:returns HostCollection object

`robottelo.cli.factory.make_job_invocation (options=None)`

Creates a Job Invocation

Parameters `options` – Check options using *hammer job-invocation create –help* on satellite.

:returns JobInvocation object

`robottelo.cli.factory.make_job_template (options=None)`

Creates a Job Template

Parameters **options** – Check options using *hammer job-template create –help* on satellite.

:returns JobTemplate object

`robottelo.cli.factory.make_user (options=None)`

Creates a User

Parameters **options** – Check options using *hammer user create –help* on satellite.

:returns User object

`robottelo.cli.factory.make_usergroup (options=None)`

Creates a User Group

Parameters **options** – Check options using *hammer user-group create –help* on satellite.

:returns UserGroup object

`robottelo.cli.factory.make_usergroup_external (options=None)`

Creates an External User Group

Parameters **options** – Check options using *hammer user-group external create –help* on satellite.

:returns UserGroupExternal object

`robottelo.cli.factory.make_ldap_auth_source (options=None)`

Creates an LDAP Auth Source

Parameters **options** – Check options using *hammer auth-source ldap create –help* on satellite.

:returns LDAPAuthSource object

`robottelo.cli.factory.make_compute_resource (options=None)`

Creates a Compute Resource

Parameters **options** – Check options using *hammer compute-resource create –help* on satellite.

:returns ComputeResource object

`robottelo.cli.factory.make_org (options=None)`

Creates an Organization

Parameters **options** – Check options using *hammer organization create –help* on satellite.

:returns Organization object

`robottelo.cli.factory.make_org_with_credentials (options=None, credentials=None)`

Helper function to create organization with credentials

`robottelo.cli.factory.make_realm (options=None)`

Creates a REALM

Parameters **options** – Check options using *hammer realm create –help* on satellite.

:returns Realm object

`robottelo.cli.factory.make_report_template (options=None)`

Creates a Report Template

Parameters **options** – Check options using *hammer report-template create –help* on satellite.

:returns ReportTemplate object

`robottelo.cli.factory.make_os (options=None)`

Creates an Operating System

Parameters options – Check options using *hammer os create –help* on satellite.

:returns OperatingSys object

`robottelo.cli.factory.make_scapcontent (options=None)`

Creates Scap Content

Parameters options – Check options using *hammer scap-content create –help* on satellite.

:returns ScapContent object

`robottelo.cli.factory.make_domain (options=None)`

Creates a Domain

Parameters options – Check options using *hammer domain create –help* on satellite.

:returns Domain object

`robottelo.cli.factory.make_hostgroup (options=None)`

Creates a Hostgroup

Parameters options – Check options using *hammer hostgroup create –help* on satellite.

:returns Hostgroup object

`robottelo.cli.factory.make_medium (options=None)`

Creates a Medium

Parameters options – Check options using *hammer medium create –help* on satellite.

:returns Medium object

`robottelo.cli.factory.make_environment (options=None)`

Creates a Puppet Environment

Parameters options – Check options using *hammer environment create –help* on satellite.

:returns Environment object

`robottelo.cli.factory.make_lifecycle_environment (options=None)`

Creates a Lifecycle Environment

Parameters options – Check options using *hammer lifecycle-environment create –help* on satellite.

:returns LifecycleEnvironment object

`robottelo.cli.factory.make_tailoringfile (options=None)`

Creates a tailoring File

Parameters options – Check options using *hammer tailoring-file create –help* on satellite.

:returns TailoringFile object

`robottelo.cli.factory.make_template (options=None)`

Creates a Template

Parameters options – Check options using *hammer template create –help* on satellite.

:returns Template object

`robottelo.cli.factory.make_template_input (options=None)`

Creates Template Input

Parameters options – Check options using *hammer template-input create –help* on satellite.

:returns TemplateInput object

`robottelo.cli.factory.make_virt_who_config (options=None)`
Creates a Virt Who Configuration

Parameters `options` – Check options using `hammer virt-who-config create --help` on satellite.

:returns VirtWhoConfig object

`robottelo.cli.factory.activationkey_add_subscription_to_repo (options=None)`
Helper function that adds subscription to an activation key

`robottelo.cli.factory.setup_org_for_a_custom_repo (options=None)`
Sets up Org for the given custom repo by:

1. **Checks if organization and lifecycle environment were given, otherwise** creates new ones.
2. Creates a new product with the custom repo. Synchronizes the repo.
3. **Checks if content view was given, otherwise creates a new one and**
 - adds the RH repo
 - publishes
 - promotes to the lifecycle environment
4. **Checks if activation key was given, otherwise creates a new one and** associates it with the content view.
5. Adds the custom repo subscription to the activation key

Returns A dictionary with the entity ids of Activation key, Content view, Lifecycle Environment, Organization, Product and Repository

`robottelo.cli.factory._setup_org_for_a_rh_repo (options=None)`
Sets up Org for the given Red Hat repository by:

1. **Checks if organization and lifecycle environment were given, otherwise** creates new ones.
2. Clones and uploads manifest.
3. Enables RH repo and synchronizes it.
4. **Checks if content view was given, otherwise creates a new one and**
 - adds the RH repo
 - publishes
 - promotes to the lifecycle environment
5. **Checks if activation key was given, otherwise creates a new one and** associates it with the content view.
6. Adds the RH repo subscription to the activation key

Note that in most cases you should use `setup_org_for_a_rh_repo` instead as it's more flexible.

Returns A dictionary with the entity ids of Activation key, Content view, Lifecycle Environment, Organization and Repository

`robottelo.cli.factory.setup_org_for_a_rh_repo (options=None, force_manifest_upload=False, force_use_cdn=False)`

Wrapper above `_setup_org_for_a_rh_repo` to use custom downstream repo instead of CDN's 'Satellite Capsule' and 'Satellite Tools' if `settings.cdn == 0` and URL for custom repositories is set in properties.

Parameters

- **options** – a dict with options to pass to function `_setup_org_for_a_rh_repo`. See its docstring for more details
- **force_use_cdn** – bool flag whether to use CDN even if there’s downstream repo available and `settings.cdn == 0`.
- **force_manifest_upload** – bool flag whether to upload a manifest to organization even if downstream custom repo is used instead of CDN. Useful when test relies on organization with manifest (e.g. uses some other RH repo afterwards). Defaults to False.

Returns a dict with entity ids (see `_setup_org_for_a_rh_repo` and `setup_org_for_a_custom_repo`).

`robottelo.cli.factory.configure_env_for_provision` (*org=None, loc=None*)

Create and configure org, loc, product, repo, env. Update proxy, domain, subnet, compute resource, provision templates and medium with previously created entities and create a hostgroup using all mentioned entities.

Parameters

- **org** – Default Organization that should be used in both host discovering and host provisioning procedures
- **loc** – Default Location that should be used in both host discovering and host provisioning procedures

Returns List of created entities that can be re-used further in provisioning or validation procedure (e.g. hostgroup or subnet)

`robottelo.cli.factory.publish_puppet_module` (*puppet_modules, repo_url, organization_id=None*)

Creates puppet repo, sync it via provided url and publish using Content View publishing mechanism. It makes puppet class available via Puppet Environment created by Content View and returns Content View entity.

Parameters

- **puppet_modules** – List of dictionaries with module ‘author’ and module ‘name’ fields.
- **repo_url** (*str*) – Url of the repo that can be synced using pulp: pulp repo or puppet forge.
- **organization_id** – Organization id that is shared between created entities.

Returns Content View entity.

`robottelo.cli.factory.setup_virtual_machine` (*vm, org_label, rh_repos_id=None, repos_label=None, product_label=None, lce=None, activation_key=None, patch_os_release_distro=None, install_katello_agent=True*)

Setup a Virtual machine with basic components and tasks.

Parameters

- **vm** (`robottelo.vm.VirtualMachine`) – The Virtual machine to setup.
- **org_label** (*str*) – The Organization label.
- **rh_repos_id** (*list*) – a list of RH repositories ids to enable.
- **repos_label** (*list*) – a list of custom repositories labels to enable.
- **product_label** (*str*) – product label if repos_label is applicable.
- **lce** (*str*) – Lifecycle environment label if applicable.

- **activation_key** (*str*) – Activation key name if applicable.
- **patch_os_release_distro** (*str*) – distro name, to patch the VM with os version.
- **install_katello_agent** (*bool*) – whether to install katello agent.

`robottelo.cli.factory._get_capsule_vm_distro_repos` (*distro*)

Return the right RH repos info for the capsule setup

`robottelo.cli.factory.add_role_permissions` (*role_id*, *resource_permissions*)

Create role permissions found in resource permissions dict

Parameters

- **role_id** – The role id
- **resource_permissions** – a dict containing resources with permission names and other Filter options

Usage:

```
role = make_role({'organization-id': org['id']})
resource_permissions = {
    'Katello::ActivationKey': {
        'permissions': [
            'view_activation_keys',
            'create_activation_keys',
            'edit_activation_keys',
            'destroy_activation_keys'
        ],
        'search': "name ~ {}".format(ak_name_like)
    },
}
add_role_permissions(role['id'], resource_permissions)
```

`robottelo.cli.factory.setup_cdn_and_custom_repositories` (*org_id*, *repos*, *download_policy='on_demand'*, *synchronize=True*)

Setup cdn and custom repositories

Parameters

- **org_id** (*int*) – The organization id
- **repos** (*list*) – a list of dict repositories options
- **download_policy** (*str*) – update the repositories with this download policy
- **synchronize** (*bool*) – Whether to synchronize the repositories.

Returns a dict containing the content view and repos info

`robottelo.cli.factory.setup_cdn_and_custom_repos_content` (*org_id*, *lce_id=None*, *repos=None*, *upload_manifest=True*, *download_policy='on_demand'*, *rh_subscriptions=None*, *default_cv=False*)

Setup cdn and custom repositories, content view and activations key

Parameters

- **org_id** (*int*) – The organization id

- **lce_id** (*int*) – the lifecycle environment id
- **repos** (*list*) – a list of dict repositories options
- **default_cv** (*bool*) – whether to use the Default Organization CV
- **upload_manifest** (*bool*) – whether to upload the organization manifest
- **download_policy** (*str*) – update the repositories with this download policy
- **rh_subscriptions** (*list*) – a list of RH subscription to attach to activation key

Returns a dict containing the activation key, content view and repos info

`robottelo.cli.factory.vm_setup_ssh_config` (*vm, ssh_key_name, host, user=None*)

Create host entry in vm ssh config and know_hosts files to allow vm to access host via ssh without password prompt

Parameters

- **vm** (`robottelo.vm.VirtualMachine`) – Virtual machine instance
- **ssh_key_name** (*str*) – The ssh key file name to use to access host, the file must already exist in /root/.ssh directory
- **host** (*str*) – the hostname to setup that will be accessed from vm
- **user** (*str*) – the user that will access the host

`robottelo.cli.factory.vm_upload_ssh_key` (*vm, source_key_path, destination_key_name*)

Copy ssh key to virtual machine ssh path and ensure proper permission is set

Parameters

- **vm** (`robottelo.vm.VirtualMachine`) – Virtual machine instance
- **source_key_path** – The ssh key file path to copy to vm
- **destination_key_name** – The ssh key file name when copied to vm

`robottelo.cli.factory.virt_who_hypervisor_config` (*config_id, org_id=None, hypervisor_hostname=None, configure_ssh=False, hypervisor_user=None, subscription_name=None, exec_one_shot=False, upload_manifest=True, extra_repos=None, virt_who_vm, lce_id=None*)

Configure virtual machine as hypervisor virt-who service

Parameters

- **config_id** (*int*) – virt-who config id
- **virt_who_vm** (`robottelo.vm.VirtualMachine`) – the Virtual machine instance to use for configuration
- **org_id** (*int*) – the organization id
- **lce_id** (*int*) – the lifecycle environment id to use
- **hypervisor_hostname** (*str*) – the hypervisor hostname
- **hypervisor_user** (*str*) – hypervisor user that connect with the ssh key

- **configure_ssh** (*bool*) – whether to configure the ssh key to allow this virtual machine to connect to hypervisor
- **subscription_name** (*str*) – the subscription name to assign to virt-who hypervisor guests
- **exec_one_shot** (*bool*) – whether to run the virt-who one-shot command after startup
- **upload_manifest** (*bool*) – whether to upload the organization manifest
- **extra_repos** (*list*) – (Optional) a list of repositories dict options to setup additionally.

`robottelo.cli.factory.make_http_proxy` (*options=None*)

Creates a HTTP Proxy

Parameters **options** – Check options using `hammer http-proxy create -help` on satellite.

:returns HttpProxy object

`robottelo.cli.file`

Usage:

```
hammer file [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

<code>info</code>	Show a file
<code>list</code>	List files

Module Contents

Classes

<code>File()</code>	Manipulates files command.
---------------------	----------------------------

class `robottelo.cli.file.File`

Bases: `robottelo.cli.base.Base`

Manipulates files command.

command_base = `file`

`robottelo.cli.filter`

Usage:: `hammer filter [OPTIONS] SUBCOMMAND [ARG] ...`

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

available-permissions	List all permissions
available-resources	List available resource types.
create	Create a filter
delete	Delete a filter
info	Show a filter
list	List all filters
update	Update a filter

Module Contents

Classes

<i>Filter()</i>	Manipulates Katello's filter command.
-----------------	---------------------------------------

class robottelo.cli.filter.**Filter**

Bases: *robottelo.cli.base.Base*

Manipulates Katello's filter command.

command_base = **filter**

classmethod **available_permissions** (*cls, options=None*)

robottelo.cli.globalparam

Usage:

hammer global -parameter [OPTIONS] SUBCOMMAND [ARG] ...
--

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

delete	Delete a common_parameter
list	List all common parameters.
set	Set a global parameter.

Module Contents

Classes

*GlobalParameter()*Manipulates Foreman's global parameters.

class robottelo.cli.globalparam.**GlobalParameter**Bases: *robottelo.cli.base.Base*

Manipulates Foreman's global parameters.

command_base = **global-parameter****classmethod** **set** (*cls, options=None*)

Set global parameter

robottelo.cli.gpgkey

Usage:

```
hammer gpg [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create a GPG Key
delete	Destroy a GPG Key
info	Show a GPG key
list	List GPG Keys
update	Update a GPG Key

Module Contents

Classes

*GPGKey()*Manipulates Foreman's GPG Keys.

class robottelo.cli.gpgkey.**GPGKey**Bases: *robottelo.cli.base.Base*

Manipulates Foreman's GPG Keys.

command_base = **gpg****command_requires_org** = **True****classmethod** **info** (*cls, options=None*)

Gets information for GPG Key

robottelo.cli.hammer

Helpers to interact with hammer command line utility.

Module Contents

Functions

<code>_csv_reader(output)</code>	An unicode CSV reader which processes unicode strings and return unicode
<code>_normalize(header)</code>	Replace empty spaces with '-' and lower all chars
<code>parse_json(stdout)</code>	Parse JSON output from Hammer CLI and convert it to python dictionary
<code>_normalize_obj(obj)</code>	Normalize all dict's keys replacing empty spaces with "-" and lowering
<code>parse_csv(output)</code>	Parse CSV output from Hammer CLI and convert it to python dictionary.
<code>parse_help(output)</code>	Parse the help output from a hammer command and return a dictionary
<code>get_line_indentation_spaces(line, tab_spaces=4)</code>	Return the number of spaces chars the line begin with
<code>get_line_indentation_level(line, tab_spaces=4, indentation_spaces=4)</code>	Return the indentation level
<code>parse_info(output)</code>	Parse the info output and returns a dict mapping the values.

`robottelo.cli.hammer._csv_reader(output)`

An unicode CSV reader which processes unicode strings and return unicode strings data.

This is needed because the builtin module does not support unicode strings, from Python 2 docs:

Note: This version of the csv module doesn't support Unicode input. Also, there are currently some issues regarding ASCII NUL characters. Accordingly, all input should be UTF-8 or printable ASCII to be safe;"

On Python 3 this generator is not needed because the default string type is unicode.

Parameters `output` – can be any object which supports the iterator protocol and returns a unicode string each time its `next()` method is called.

Returns generator that will yield a list of unicode string values.

`robottelo.cli.hammer._normalize(header)`

Replace empty spaces with '-' and lower all chars

`robottelo.cli.hammer.parse_json(stdout)`

Parse JSON output from Hammer CLI and convert it to python dictionary while normalizing keys.

`robottelo.cli.hammer._normalize_obj(obj)`

Normalize all dict's keys replacing empty spaces with "-" and lowering chars

`robottelo.cli.hammer.parse_csv(output)`

Parse CSV output from Hammer CLI and convert it to python dictionary.

`robottelo.cli.hammer.parse_help(output)`

Parse the help output from a hammer command and return a dictionary mapping the subcommands and options accepted by that command.

`robottelo.cli.hammer.get_line_indentation_spaces(line, tab_spaces=4)`

Return the number of spaces chars the line begin with

Parameters

- **line** (*str*) – the line string to parse
- **tab_spaces** (*int*) – The tab char is represent how many spaces

```
robottelo.cli.hammer.get_line_indentation_level(line, tab_spaces=4, indentation_spaces=4)
```

Return the indentation level

Parameters

- **line** (*str*) – the line string to parse
- **tab_spaces** (*int*) – The tab char is represent how many spaces
- **indentation_spaces** – how much spaces represent an indentation level

Note:

```
suppose we have the following lines:
'''
level 0
    level 1
        level 2
'''
assert get_line_indentation_level('level 0') == 0
assert get_line_indentation_level('    level 1') == 1
assert get_line_indentation_level('        level 2') == 2
```

```
robottelo.cli.hammer.parse_info(output)
```

Parse the info output and returns a dict mapping the values.

robottelo.cli.host

Usage:

```
hammer host [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

ansible-roles	Manage Ansible roles on a host
boot	Boot host from specified device
config-reports	List all reports
create	Create a host
delete	Delete a host
delete-parameter	Delete parameter for a host
disassociate	Disassociate a host
enc-dump	Dump host's ENC YAML
errata	Manage errata on your hosts
facts	List all fact values
info	Show a host
interface	View and manage host's network interfaces
list	List all hosts
package	Manage packages on your hosts
package-group	Manage package-groups on your hosts

(continues on next page)

(continued from previous page)

policies-enc	View policies ENC for host
puppet-classes	List all Puppet classes
puppetrun	Force a Puppet agent run on the host
reboot	Reboot a host
rebuild-config	Rebuild orchestration related configurations for host
reports	List all reports
reset	Reset a host
sc-params	List all smart class parameters
set-parameter	Create or append a parameter for a host
start	Power a host on
status	Get status of host
stop	Power a host off
subscription	Manage subscription information on your hosts
update	Update a host

Module Contents

Classes

<code>Host()</code>	Manipulates Foreman's hosts.
<code>HostInterface()</code>	Manages interface functionality for hosts.

class `robottelo.cli.host.Host`

Bases: `robottelo.cli.base.Base`

Manipulates Foreman's hosts.

command_base = `host`

classmethod `ansible_roles_play` (*cls*, *options*)

Plays the associated ansible-roles

classmethod `enc_dump` (*cls*, *options*)

Dump host's ENC YAML.

Usage:

```
hammer host enc-dump [OPTIONS]
```

Options:

```
--id ID
--location LOCATION_NAME          Location name
--location-id LOCATION_ID
--location-title LOCATION_TITLE  Location title
--name NAME                       Host name
--organization ORGANIZATION_NAME Organization name
--organization-id ORGANIZATION_ID Organization ID
--organization-title ORGANIZATION_TITLE Organization title
-h, --help                        Print help
```

classmethod `errata_apply` (*cls*, *options*)

Schedule errata for installation

classmethod errata_info (*cls, options*)

Retrieve a single errata for a system

classmethod errata_list (*cls, options*)

List errata available for the content host.

classmethod facts (*cls, options=None*)

List all fact values.

Usage:

```
hammer host facts [OPTIONS]
```

Options:

```
--id ID                resource id
--name NAME             resource name
--order ORDER          sort results
--page PAGE            paginate results
--per-page PER_PAGE    number of entries per request
--search SEARCH        filter results
-h, --help             print help
```

classmethod package_install (*cls, options*)

Install packages remotely.

classmethod package_list (*cls, options*)

List packages installed on the host.

classmethod package_remove (*cls, options*)

Uninstall packages remotely.

classmethod package_upgrade (*cls, options*)

Update packages remotely.

classmethod package_upgrade_all (*cls, options*)

Update all packages remotely.

classmethod package_group_install (*cls, options*)

Install package groups remotely.

classmethod package_group_remove (*cls, options*)

Uninstall package groups remotely.

classmethod puppetrun (*cls, options=None*)

Force a puppet run on the agent.

Usage:

```
hammer host puppetrun [OPTIONS]
```

Options:

```
--id ID                resource id
--name NAME             resource name
-h, --help             print help
```

classmethod reboot (*cls, options=None*)

Reboot a host

Usage:

```
hammer host reboot [OPTIONS]
```

Options:

```
--id ID                resource id
--name NAME            resource name
-h, --help            print help
```

classmethod reports (*cls, options=None*)

List all reports.

Usage:

```
hammer host reports [OPTIONS]
```

Options:

```
--id ID                resource id
--name NAME            resource name
--order ORDER          sort results
--page PAGE            paginate results
--per-page PER_PAGE    number of entries per request
--search SEARCH        filter results
-h, --help            print help
```

classmethod start (*cls, options=None*)

Power a host on

Usage:

```
hammer host start [OPTIONS]
```

Options:

```
--id ID                resource id
--name NAME            resource name
-h, --help            print help
```

classmethod status (*cls, options=None*)

Get status of host

Usage:

```
hammer host status [OPTIONS]
```

Options:

```
--id ID                resource id
--name NAME            resource name
-h, --help            print help
```

classmethod stop (*cls, options=None*)

Power a host off

Usage:

```
hammer host stop [OPTIONS]
```

Options:

```

--force                Force turning off a host
--id ID                resource id
--name NAME            resource name
-h, --help             print help

```

classmethod `subscription_register` (*cls, options=None*)

Register a host with subscription and information.

Usage:

```
hammer host subscription register [OPTIONS]
```

Options:

```

--content-view CONTENT_VIEW_NAME      Content view
                                       name to search
                                       by
--content-view-id CONTENT_VIEW_ID     content view
                                       numeric
                                       identifier
--hypervisor-guest-uuids HYPERVISOR_GUEST_UUIDS  UUIDs of the
                                       virtual guests
                                       from the
                                       host's
                                       hypervisor
                                       Comma separated
                                       list of values.
--lifecycle-environment LIFECYCLE_ENVIRONMENT_NAME  Lifecycle
                                       environment
                                       name to search
                                       by
--lifecycle-environment-id LIFECYCLE_ENVIRONMENT_ID  ID of the
                                       environment
--name NAME                             Name of the
                                       host
--organization ORGANIZATION_NAME       Organization
                                       name to search
                                       by
--organization-id ORGANIZATION_ID      organization ID
--organization-label ORGANIZATION_LABEL  Organization
                                       label to search
                                       by
--release-version RELEASE_VERSION      Release version
                                       of the content
                                       host
--service-level SERVICE_LEVEL           A service level
                                       for
                                       auto-healing
                                       process, e.g.
                                       SELF-SUPPORT
--uuid UUID                              UUID to use for
                                       registered
                                       host, random
                                       uuid is
                                       generated if
                                       not provided

```

classmethod `subscription_unregister` (*cls, options=None*)

Unregister the host as a subscription consumer.

Usage:

```
hammer host subscription unregister [OPTIONS]
```

Options:

```
--host HOST_NAME          Name to search by
--host-id HOST_ID         Host ID
```

classmethod subscription_attach (*cls, options=None*)

Attach a subscription to host

Usage:

```
hammer host subscription attach [OPTIONS]
```

Options:

```
--host HOST_NAME          Name to search by
--host-id HOST_ID         Host ID
--quantity Quantity      Quantity of this subscriptions to
                           add. Defaults to 1
--subscription-id SUBSCRIPTION_ID ID of subscription
```

classmethod subscription_remove (*cls, options=None*)

Remove a subscription from host

Usage:

```
hammer host subscription remove [OPTIONS]
```

Options:

```
--host HOST_NAME          Name to search by
--host-id HOST_ID         Host ID
--quantity Quantity      Remove the first instance of a
                           subscription with matching id
                           and quantity
--subscription-id SUBSCRIPTION_ID ID of subscription
```

classmethod subscription_auto_attach (*cls, options=None*)

Auto attach subscription to host

Usage:

```
hammer host subscription auto-attach [OPTIONS]
```

Options:

```
--host HOST_NAME          Name to search by
--host-id HOST_ID         Host ID
-h, --help                print help
```

classmethod sc_params (*cls, options=None*)

List all smart class parameters

Usage:

```
hammer host sc-params [OPTIONS]
```

Options:

```
--host HOST_NAME           Host name
--host-id HOST_ID          Host ID
--order ORDER              sort results
--page PAGE                paginate results
--per-page PER_PAGE        number of entries per request
--search SEARCH            filter results
```

class robottelo.cli.host.**HostInterface**Bases: *robottelo.cli.base.Base*

Manages interface functionality for hosts.

Usage:: hammer host interface [OPTIONS] SUBCOMMAND [ARG] ...**Subcommands::** create Create an interface on a host delete Delete a host's interface info Show an interface for host list List all interfaces for host update Update a host's interface**command_base = host interface****classmethod create** (*cls, options=None*)

Create new network interface for host

robottelo.cli.hostcollection

Usage:

```
hammer host-collection [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

add-host	Add host to the host collection
copy	Make copy of a host collection
create	Create a host collection
delete	Destroy a host collection
erratum	Manipulate errata for a host collection
hosts	List all hosts
info	Show a host collection
list	List host collections
package	Manipulate packages for a host collection
package-group	Manipulate package-groups for a host collection
remove-host	Remove hosts from the host collection
update	Update a host collection

Module Contents

Classes

*HostCollection()*Manipulates Katello engine's host-collection command.

class robottelo.cli.hostcollection.**HostCollection**Bases: *robottelo.cli.base.Base*

Manipulates Katello engine's host-collection command.

command_base = **host-collection****classmethod** **add_host** (*cls, options=None*)

Add host to the host collection

classmethod **remove_host** (*cls, options=None*)

Remove hosts from the host collection

classmethod **hosts** (*cls, options=None*)

List hosts added to the host collection

Usage:

```
hammer host-collection hosts [OPTIONS]
```

Options:

```
--environment ENVIRONMENT_NAME      Name to search by
--environment-id ENVIRONMENT_ID
--hostgroup HOSTGROUP_NAME           Name to search by
--hostgroup-id HOSTGROUP_ID
--id HOST_COLLECTION_ID              Host Collection ID
--location LOCATION_NAME             Name to search by
--location-id LOCATION_ID
--name HOST_COLLECTION_NAME          Host Collection Name
--order ORDER                        sort results
--organization ORGANIZATION_NAME     Organization name to
                                     search by
--organization-id ORGANIZATION_ID    organization ID
--organization-label ORGANIZATION_LABEL Organization label to
                                     search by
--page PAGE                          paginate results
--per-page PER_PAGE                  number of entries per
                                     request
--search SEARCH                      filter results
-h, --help                           print help
```

classmethod **erratum_install** (*cls, options*)

Schedule errata for installation

classmethod **package_install** (*cls, options*)

Schedule package for installation

classmethod **copy** (*cls, options*)

Clone existing host collection

robottelo.cli.hostgroup

Usage:

```
hammer hostgroup [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

ansible-roles	Manage Ansible roles on a hostgroup
create	Create a host group
delete	Delete a host group
delete-parameter	Delete parameter for a hostgroup
info	Show a host group
list	List all host groups
puppet-classes	List all Puppet classes
rebuild-config	Rebuild orchestration config
sc-params	List all smart class parameters
set-parameter	Create or update parameter for a hostgroup
update	Update a host group

Module Contents**Classes***HostGroup()*

Manipulates Foreman's hostgroups.

class robottelo.cli.hostgroup.**HostGroup**Bases: *robottelo.cli.base.Base*

Manipulates Foreman's hostgroups.

command_base = **hostgroup****classmethod** **sc_params** (*cls, options=None*)

List all smart class parameters

Usage:

```
hammer hostgroup sc-params [OPTIONS]
```

Options:

--hostgroup HOSTGROUP_NAME	Hostgroup name
--hostgroup-id HOSTGROUP_ID	
--hostgroup-title HOSTGROUP_TITLE	Hostgroup title
--order ORDER	sort results
--page PAGE	paginate results
--per-page PER_PAGE	number of entries per request
--search SEARCH	filter results

robottelo.cli.http_proxy**Usage:** http-proxy [OPTIONS] SUBCOMMAND [ARG] ...

Parameters: SUBCOMMAND Subcommand [ARG] ... Subcommand arguments

Subcommands: create Create an HTTP Proxy delete Delete an HTTP Proxy info Show an HTTP Proxy list List of HTTP Proxies update Update an HTTP Proxy

Options:

-h, --help Print help

Module Contents

Classes

<i>HttpProxy()</i>	Manipulates http-proxy command.
--------------------	---------------------------------

class robottelo.cli.http_proxy.**HttpProxy**

Bases: *robottelo.cli.base.Base*

Manipulates http-proxy command.

command_base = http-proxy

robottelo.cli.job_invocation

Usage: hammer job-invocation [OPTIONS] SUBCOMMAND [ARG] ...

Parameters: SUBCOMMAND subcommand [ARG] ... subcommand arguments

Subcommands:

create Create a job invocation info Show job invocation list List job invocations output View the output for a host

Module Contents

Classes

<i>JobInvocation()</i>	Run remote jobs.
------------------------	------------------

class robottelo.cli.job_invocation.**JobInvocation**

Bases: *robottelo.cli.base.Base*

Run remote jobs.

command_base = job-invocation

classmethod **get_output** (*cls, options*)

Get output of the job invocation

robottelo.cli.job_template

Usage: hammer job-template [OPTIONS] SUBCOMMAND [ARG] ...

Parameters:

SUBCOMMAND subcommand [ARG] ... subcommand arguments

Subcommands:

create Create a job template delete Delete a job template dump View job template content info Show job template details list List job templates update Update a job template

Module Contents**Classes**

JobTemplate()

Manipulate job templates.

class robottelo.cli.job_template.**JobTemplate**

Bases: *robottelo.cli.base.Base*

Manipulate job templates.

command_base = job-template

robottelo.cli.ldapauthsource

Usage:

```
hammer auth-source ldap [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:: create Create an LDAP authentication source delete Delete an LDAP authentication source info Show an LDAP authentication source list List all LDAP authentication sources update Update an LDAP authentication source

Module Contents**Classes**

LDAPAuthSource()

Manipulates LDAP auth source

class robottelo.cli.ldapauthsource.**LDAPAuthSource**

Bases: *robottelo.cli.base.Base*

Manipulates LDAP auth source

command_base = auth-source ldap

robottelo.cli.lifecycleenvironment

Usage:

```
hammer lifecycle-environment [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

<code>list</code>	List environments in an organization
<code>update</code>	Update an environment
<code>create</code>	Create an environment
<code>delete</code>	Destroy an environment
<code>info</code>	Show an environment

Module Contents

Classes

<code>LifecycleEnvironment()</code>	Manipulates Katello engine's lifecycle-environment command.
-------------------------------------	---

class robottelo.cli.lifecycleenvironment.LifecycleEnvironment

Bases: `robottelo.cli.base.Base`

Manipulates Katello engine's lifecycle-environment command.

command_base = `lifecycle-environment`

command_requires_org = `True`

classmethod list (*cls, options=None, per_page=False*)

List information. @param options: ID (sometimes name works as well) to retrieve info.

classmethod paths (*cls, options=None*)

robottelo.cli.location

Usage:

```
hammer location [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

add-compute-resource	Associate a compute resource
add-domain	Associate a domain
add-environment	Associate an environment
add-hostgroup	Associate a hostgroup
add-medium	Associate a medium
add-organization	Associate an organization
add-provisioning-template	Associate provisioning templates
add-smart-proxy	Associate a smart proxy
add-subnet	Associate a subnet
add-user	Associate an user
create	Create a location
delete	Delete a location
info	Show a location
list	List <i>all</i> locations
remove-compute-resource	Disassociate a compute resource
remove-domain	Disassociate a domain
remove-environment	Disassociate an environment
remove-hostgroup	Disassociate a hostgroup
remove-medium	Disassociate a medium
remove-organization	Disassociate an organization
remove-provisioning-template	Disassociate provisioning templates
remove-smart-proxy	Disassociate a smart proxy
remove-subnet	Disassociate a subnet
remove-user	Disassociate an user
update	Update a location

Module Contents

Classes

Location()

Manipulates Foreman's Locations

class robottelo.cli.location.**Location**

Bases: *robottelo.cli.base.Base*

Manipulates Foreman's Locations

command_base = **location**

classmethod **add_compute_resource** (*cls, options=None*)

Associate a compute resource

classmethod **add_domain** (*cls, options=None*)

Associate a domain

classmethod **add_environment** (*cls, options=None*)

Associate an environment

classmethod **add_hostgroup** (*cls, options=None*)

Associate a hostgroup

classmethod **add_medium** (*cls, options=None*)

Associate a medium

classmethod **add_organization** (*cls, options=None*)

Associate an organization

- classmethod add_provisioning_template** (*cls, options=None*)
Associate a provisioning template
- classmethod add_smart_proxy** (*cls, options=None*)
Associate a smart proxy
- classmethod add_subnet** (*cls, options=None*)
Associate a subnet
- classmethod add_user** (*cls, options=None*)
Associate a user
- classmethod remove_compute_resource** (*cls, options=None*)
Disassociate a compute resource
- classmethod remove_domain** (*cls, options=None*)
Disassociate a domain
- classmethod remove_environment** (*cls, options=None*)
Disassociate an environment
- classmethod remove_hostgroup** (*cls, options=None*)
Disassociate a hostgroup
- classmethod remove_medium** (*cls, options=None*)
Disassociate a medium
- classmethod remove_organization** (*cls, options=None*)
Disassociate an organization
- classmethod remove_provisioning_template** (*cls, options=None*)
Disassociate a provisioning template
- classmethod remove_smart_proxy** (*cls, options=None*)
Disassociate a smart proxy
- classmethod remove_subnet** (*cls, options=None*)
Disassociate a subnet
- classmethod remove_user** (*cls, options=None*)
Disassociate a user

robottelo.cli.medium

Usage:

```
hammer medium [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

add_operatingsystem	Associate a resource
create	Create a medium.
delete	Delete a medium.
info	Show a medium.
list	List <i>all</i> media.

(continues on next page)

(continued from previous page)

remove_operatingsystem	Disassociate a resource
update	Update a medium.

Module Contents

Classes

<i>Medium()</i>	Manipulates Foreman's installation media.
-----------------	---

```
class robottelo.cli.medium.Medium
    Bases: robottelo.cli.base.Base
    Manipulates Foreman's installation media.
    command_base = medium
```

`robottelo.cli.model`

Usage:

```
hammer model [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create a model.
delete	Delete a model.
info	Show a model.
list	List all models.
update	Update a model.

Module Contents

Classes

<i>Model()</i>	Manipulates Foreman's hardware model.
----------------	---------------------------------------

```
class robottelo.cli.model.Model
    Bases: robottelo.cli.base.Base
    Manipulates Foreman's hardware model.
    command_base = model
```

robottelo.cli.module_stream

Usage:

```
hammer module-stream [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

info	Show a module-stream
list	List module-streams

Module Contents

Classes

<i>ModuleStream()</i>	Manipulates module-stream command.
-----------------------	------------------------------------

class robottelo.cli.module_stream.**ModuleStream**

Bases: *robottelo.cli.base.Base*

Manipulates module-stream command.

command_base = module-stream

robottelo.cli.operatingsys

Usage:

```
hammer os [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

add-architecture	Associate a resource
add-provisioning-template	Associate provisioning templates
add-ptable	Associate a resource
create	Create an OS.
delete	Delete an OS.
delete-default-template	
delete-parameter	Delete parameter for an operating system.
info	Show an OS.
list	List all operating systems.
remove-architecture	Disassociate a resource
remove-provisioning-template	Disassociate provisioning templates

(continues on next page)

(continued from previous page)

remove-ptable	Disassociate a resource
set-default-template	
set-parameter	Create or update parameter for an operating system.
update	Update an OS.

Module Contents

Classes

OperatingSys()

Manipulates Foreman's operating systems.

class `robottelo.cli.operatingsys.OperatingSys`

Bases: `robottelo.cli.base.Base`

Manipulates Foreman's operating systems.

command_base = `os`

classmethod `add_architecture` (*cls*, *options=None*)

Adds existing architecture to OS.

classmethod `add_provisioning_template` (*cls*, *options=None*)

Adds existing template to OS.

classmethod `add_ptable` (*cls*, *options=None*)

Adds existing partitioning table to OS.

classmethod `remove_architecture` (*cls*, *options=None*)

Removes architecture from OS.

classmethod `remove_provisioning_template` (*cls*, *options=None*)

Removes template from OS.

classmethod `remove_ptable` (*cls*, *options=None*)

Removes partitioning table from OS.

`robottelo.cli.org`

Usage:

```
hammer organization [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

add-computeresource	Associate a resource
add-domain	Associate a resource
add-environment	Associate a resource
add-hostgroup	Associate a resource

(continues on next page)

(continued from previous page)

add-location	Associate a location
add-medium	Associate a resource
add-provisioning-template	Associate provisioning templates
add-smartproxy	Associate a resource
add-subnet	Associate a resource
add-user	Associate a resource
create	Create an organization
delete	Delete an organization
delete-parameter	Delete parameter for an organization.
info	Show an organization
list	List all organizations
remove_computeresource	Disassociate a resource
remove_domain	Disassociate a resource
remove_environment	Disassociate a resource
remove_hostgroup	Disassociate a resource
remove-location	Disassociate a location
remove_medium	Disassociate a resource
remove-provisioning-template	Disassociate provisioning templates
remove_smartproxy	Disassociate a resource
remove_subnet	Disassociate a resource
remove_user	Disassociate a resource
set-parameter	Create or update parameter for an organization.
update	Update an organization

Module Contents

Classes

Org()

Manipulates Foreman's Organizations

class robottelo.cli.org.Org

Bases: *robottelo.cli.base.Base*

Manipulates Foreman's Organizations

command_base = organization

classmethod add_compute_resource (*cls, options=None*)

Adds a computeresource to an org

classmethod remove_compute_resource (*cls, options=None*)

Removes a computeresource from an org

classmethod add_domain (*cls, options=None*)

Adds a domain to an org

classmethod remove_domain (*cls, options=None*)

Removes a domain from an org

classmethod add_environment (*cls, options=None*)

Adds an environment to an org

classmethod remove_environment (*cls, options=None*)

Removes an environment from an org

classmethod add_hostgroup (*cls, options=None*)
Adds a hostgroup to an org

classmethod remove_hostgroup (*cls, options=None*)
Removes a hostgroup from an org

classmethod add_location (*cls, options=None*)
Adds a location to an org

classmethod remove_location (*cls, options=None*)
Removes a location from an org

classmethod add_medium (*cls, options=None*)
Adds a medium to an org

classmethod remove_medium (*cls, options=None*)
Removes a medium from an org

classmethod add_provisioning_template (*cls, options=None*)
Adds a provisioning template to an org

classmethod remove_provisioning_template (*cls, options=None*)
Removes a provisioning template from an org

classmethod add_smart_proxy (*cls, options=None*)
Adds a smartproxy to an org

classmethod remove_smart_proxy (*cls, options=None*)
Removes a smartproxy from an org

classmethod add_subnet (*cls, options=None*)
Adds existing subnet to an org

classmethod remove_subnet (*cls, options=None*)
Removes a subnet from an org

classmethod add_user (*cls, options=None*)
Adds an user to an org

classmethod remove_user (*cls, options=None*)
Removes an user from an org

robottelo.cli.ostreebranch

Usage:

```
hammer ostree-branch [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

info	Show an ostree branch
list	List ostree_branches

Module Contents

Classes

<i>OstreeBranch()</i>	Manipulates Ostree branches.
-----------------------	------------------------------

class `robottelo.cli.ostreebranch.OstreeBranch`

Bases: `robottelo.cli.base.Base`

Manipulates Ostree branches.

`command_base = ostree-branch`

`robottelo.cli.package`

Usage:

```
hammer package [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

<code>info</code>	Show a package
<code>list</code>	List packages

Module Contents

Classes

<i>Package()</i>	Manipulates packages command.
------------------	-------------------------------

class `robottelo.cli.package.Package`

Bases: `robottelo.cli.base.Base`

Manipulates packages command.

`command_base = package`

`robottelo.cli.partitiontable`

Usage:

```
hammer partition-table [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

add_operatingsystem	Associate a resource
create	Create a ptable.
delete	Delete a ptable.
dump	View partition table content.
info	Show a ptable.
list	List all ptables.
remove_operatingsystem	Disassociate a resource
update	Update a ptable.

Module Contents**Classes**

<i>PartitionTable()</i>	Manipulates Foreman's partition tables.
-------------------------	---

class robottelo.cli.partitiontable.PartitionTable

Bases: *robottelo.cli.base.Base*

Manipulates Foreman's partition tables.

command_base = partition-table

robottelo.cli.product

Usage:

hammer product [OPTIONS] SUBCOMMAND [ARG] ...

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create a product
delete	Destroy a product
info	Show a product
list	List products in an environment
remove_sync_plan	Delete assignment sync plan and product.
set_sync_plan	Assign sync plan to product.
synchronize	Sync a repository
update	Update a product
update-proxy	Updates an HTTP Proxy for a product

Module Contents

Classes

<i>Product()</i>	Manipulates Katello engine's product command.
------------------	---

```

class robottelo.cli.product.Product
    Bases: robottelo.cli.base.Base
    Manipulates Katello engine's product command.
    command_base = product
    command_requires_org = True
    classmethod remove_sync_plan (cls, options=None)
        Delete assignment sync plan and product.
    classmethod set_sync_plan (cls, options=None)
        Assign sync plan to product.
    classmethod synchronize (cls, options=None)
        Synchronize a product.
    classmethod update_proxy (cls, options=None)
        Assign Http Proxy to products.
    
```

robottelo.cli.proxy

Usage:

```
hammer proxy [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create a smart proxy.
delete	Delete a smart_proxy.
import_classes	Import puppet classes from puppet proxy.
info	Show a smart proxy.
list	List all smart_proxies.
refresh-features	Refresh smart proxy features
update	Update a smart proxy.

Module Contents

Classes

<i>Proxy()</i>	Manipulates Foreman's smart proxies.
----------------	--------------------------------------

exception `robottelo.cli.proxy.CapsuleTunnelError`

Bases: `Exception`

Raised when tunnel creation fails.

class `robottelo.cli.proxy.Proxy`

Bases: `robottelo.cli.base.Base`

Manipulates Foreman's smart proxies.

command_base = `proxy`

classmethod `import_classes` (*cls*, *options=None*)

Import puppet classes from puppet proxy.

classmethod `refresh_features` (*cls*, *options=None*)

Refreshes smart proxy features

`robottelo.cli.puppet`

Usage:

```
hammer puppet-class [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

<code>info</code>	Show a puppetclass
<code>list</code>	List all puppetclasses.
<code>sc-params</code>	List all smart class parameters

Module Contents

Classes

`Puppet()`

Search Foreman's puppet modules.

class `robottelo.cli.puppet.Puppet`

Bases: `robottelo.cli.base.Base`

Search Foreman's puppet modules.

command_base = `puppet-class`

classmethod `sc_params` (*cls*, *options=None*)

Usage: `hammer puppet-class sc-params [OPTIONS]`

Options:

--order ORDER	sort results
--page PAGE	paginate results

--per-page *PER_PAGE* number of entries per request
--puppet-class *PUPPET_CLASS_NAME* Puppet class name
--puppet-class-id *PUPPET_CLASS_ID* ID of Puppet class
--search *SEARCH* filter results

robottelo.cli.puppetmodule

Usage:

```
hammer puppet-module [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

info	Show a puppet module
list	List puppet modules

Module Contents

Classes

<i>PuppetModule</i> ()	To list OR show puppet modules.
------------------------	---------------------------------

class robottelo.cli.puppetmodule.**PuppetModule**

Bases: *robottelo.cli.base.Base*

To list OR show puppet modules.

command_base = puppet-module

robottelo.cli.realm

Usage: hammer realm [OPTIONS] SUBCOMMAND [ARG] ...

Parameters: SUBCOMMAND subcommand [ARG] ... subcommand arguments

Subcommands: create Create a realm delete Delete a realm info Show a realm list List of realms update Update a realm

Options:

-h, --help print help

Module Contents

Classes

<i>Realm()</i>	Manipulates Realm subcommand
----------------	------------------------------

```
class robottelo.cli.realm.Realm
    Bases: robottelo.cli.base.Base
    Manipulates Realm subcommand
    command_base = realm
```

```
robottelo.cli.recurring_logic
```

Usage: hammer recurring-logic [OPTIONS] SUBCOMMAND [ARG] ...

Parameters: SUBCOMMAND subcommand [ARG] ... subcommand arguments

Subcommands: cancel Cancel recurring logic info Show recurring logic details list List recurring logics

Module Contents

Classes

<i>RecurringLogic()</i>	Manipulate recurring logics
-------------------------	-----------------------------

```
class robottelo.cli.recurring_logic.RecurringLogic
    Bases: robottelo.cli.base.Base
    Manipulate recurring logics
    command_base = recurring-logic
```

```
robottelo.cli.report
```

Usage:

```
hammer report [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

delete	Delete report.
info	Show info for report.
list	List reports.

Module Contents

Classes

<i>Report()</i>	Manipulates Foreman's reports.
-----------------	--------------------------------

class robottelo.cli.report.**Report**

Bases: *robottelo.cli.base.Base*

Manipulates Foreman's reports.

command_base = report

robottelo.cli.report_template

Usage:

```
hammer report-template [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	Subcommand
[ARG] ...	Subcommand arguments

Subcommands:

clone	Clone a template
create	Create a report template
delete	Delete a report template
dump	View report content
generate	Generate report
info	Show a report template
list	List <i>all</i> report templates
report-data	Downloads a generated report
schedule	Schedule generating of a report
update	Update a report template

Module Contents

Classes

<i>ReportTemplate()</i>	Manipulates with Report Template
-------------------------	----------------------------------

class robottelo.cli.report_template.**ReportTemplate**

Bases: *robottelo.cli.base.Base*

Manipulates with Report Template

command_base = report-template

classmethod **create** (*cls, options=None*)

Creates a new record using the arguments passed via dictionary.

classmethod **generate** (*cls, options=None*)

Generate a report

classmethod clone (*cls, options=None*)

Clone a report template

classmethod report_data (*cls, options=None*)

Downloads a generated report

classmethod schedule (*cls, options=None*)

Schedule generating of a report

robottelo.cli.repository

Usage:

```
hammer repository [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create a repository
delete	Destroy a repository
export	Export a repository
info	Show a repository
list	List of repositories
remove-content	Remove content from the repository
synchronize	Sync a repository
update	Update a repository
upload-content	Upload content into the repository

Module Contents

Classes

Repository()

Manipulates Katello engine's repository command.

class robottelo.cli.repository.**Repository**

Bases: *robottelo.cli.base.Base*

Manipulates Katello engine's repository command.

command_base = repository

command_requires_org = True

classmethod create (*cls, options=None*)

Create a custom repository

classmethod export (*cls, options=None*)

Export a repository

classmethod info (*cls, options=None*)

Show a custom repository

classmethod synchronize (*cls, options, return_raw_response=None, timeout=3600*)

Synchronizes a repository.

classmethod remove_content (*cls, options*)

Remove content from a repository

classmethod upload_content (*cls, options*)

Upload content to repository.

robottelo.cli.repository_set

Implementing the repository-set hammer command

Usage:

```
hammer repository-set [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

available-repositories	Get list or available repositories for the repository set
disable	Disable a repository
enable	Enable a repository
info	Show a repository
list	List of repositories

Module Contents

Classes

RepositorySet()

Manipulates Katello engine's repository command.

class robottelo.cli.repository_set.**RepositorySet**

Bases: *robottelo.cli.base.Base*

Manipulates Katello engine's repository command.

command_base = repository-set

classmethod enable (*cls, options*)

Enables a repository.

classmethod disable (*cls, options*)

Disables a repository.

classmethod available_repositories (*cls, options*)

Lists the available repositories.

hammer repository-set available-repositories --help

Usage:

```
hammer repository-set available-repositories [OPTIONS]
```

Options:

```
--id ID                ID of the repository set
--name NAME            Repository set
                       name to search by
--organization ORGANIZATION_NAME  Organization
                       name to search by
--organization-id ORGANIZATION_ID  organization ID
--organization-label ORGANIZATION_LABEL  Organization label
                                       to search by
--product PRODUCT_NAME  Product name
                       to search by
--product-id PRODUCT_ID  product numeric identifier
-h, --help             print help
```

robottelo.cli.role**Usage:**

```
hammer role [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

clone	Clone a role
create	Create an role.
delete	Delete an role.
filters	List all filters.
info	Show a role
list	List all roles.
update	Update an role.

Module Contents**Classes**

Role()

Manipulates Katello engine's role command.

class robottelo.cli.role.**Role**

Bases: *robottelo.cli.base.Base*

Manipulates Katello engine's role command.

command_base = **role**

classmethod **filters** (*cls, options=None*)

List all filters

classmethod clone (*cls, options*)
Clone a role

robottelo.cli.scap_policy

Usage:

```
policy [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create a Policy
delete	Delete a Policy
info	Show a Policy
list	List Policies
update	Update a Policy

Module Contents

Classes

<i>Scappolicy()</i>	Manipulates Satellite's oscap policy.
---------------------	---------------------------------------

class robottelo.cli.scap_policy.**Scappolicy**

Bases: *robottelo.cli.base.Base*

Manipulates Satellite's oscap policy.

command_base = policy

robottelo.cli.scap_tailoring_files

Usage:

```
tailoring-file [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create a Tailoring file
delete	Deletes a Tailoring file
download	Show a Tailoring file as XML
info	Show a Tailoring file

(continues on next page)

(continued from previous page)

<code>list</code>	List Tailoring files
<code>update</code>	Update a Tailoring file

Module Contents

Classes

<code>TailoringFiles()</code>	Manipulates Satellite's tailoring-file.
-------------------------------	---

class `robottelo.cli.scap_tailoring_files.TailoringFiles`

Bases: `robottelo.cli.base.Base`

Manipulates Satellite's tailoring-file.

command_base = `tailoring-file`

classmethod `download_tailoring_file` (*cls, options*)

Downloads the tailoring file from satellite

`robottelo.cli.scapcontent`

Usage:

<code>scap-content [OPTIONS] SUBCOMMAND [ARG] ...</code>
--

Parameters:

<code>SUBCOMMAND</code>	subcommand
<code>[ARG] ...</code>	subcommand arguments

Subcommands:

<code>create</code>	Create SCAP content
<code>delete</code>	Deletes an SCAP content
<code>info</code>	Show an SCAP content
<code>list</code>	List SCAP contents
<code>update</code>	Update an SCAP content

Module Contents

Classes

<code>Scapcontent()</code>	Manipulates Satellite's scap-content.
----------------------------	---------------------------------------

class `robottelo.cli.scapcontent.Scapcontent`

Bases: `robottelo.cli.base.Base`

Manipulates Satellite's scap-content.

command_base = `scap-content`

robottelo.cli.sparams

Usage:

```
hammer sc-param [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

add-override-value	Create an override value for a specific smart variable
info	Show a smart class parameter
list	List all smart class parameters
remove-override-value	Delete an override value for a specific smart variable
update	Update a smart class parameter

Module Contents

Classes

SmartClassParameter()

Manipulates smart class parameters

class robottelo.cli.sparams.**SmartClassParameter**

Bases: *robottelo.cli.base.Base*

Manipulates smart class parameters

command_base = **sc-param**

classmethod **info** (*cls, options=None*)

Gets information for smart class parameter

classmethod **add_matcher** (*cls, options=None*)

Create a matcher for a specific smart class parameter

Usage:

```
hammer sc-param add-matcher [OPTIONS]
```

Options:

--location [-id -title]	Name/Title/Id of associated location
--match MATCH	Override match
--omit OMIT	Satellite will not send this parameter in classificationoutput
	One of true/false, yes/no, 1/0.
--organization [-id -title]	Name/Title/Id of associated organization
--puppet-class [-id]	Name/Id of associated puppetclass
--smart-class-parameter [-id]	Name/Id of associated smart class parameter
--value VALUE	Override value, required if omit is false

classmethod `remove_matcher` (*cls*, *options=None*)

Delete a matcher for a specific smart class parameter

Usage:

```
hammer sc-param remove-matcher [OPTIONS]
```

Options:

```
--id ID
--location[-id|-title]      Name/Title/Id of associated location
--organization[-id|-title]  Name/Title/Id of associated organization
--puppet-class[-id]        Name/Id of associated puppetclass
--smart-class-parameter[-id] Name/Id of associated smart class parameter
```

robottelo.cli.settings

Usage:

```
hammer settings [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

<code>list</code>	List all settings
<code>set</code>	Update a setting

Module Contents

Classes

`Settings()`

Manipulates Foreman's settings.

class `robottelo.cli.settings.Settings`

Bases: `robottelo.cli.base.Base`

Manipulates Foreman's settings.

command_base = `settings`

classmethod `set` (*cls*, *options=None*)

Update a setting

robottelo.cli.srpm

Usage: `hammer srpm [OPTIONS] SUBCOMMAND [ARG] ...`

Parameters: SUBCOMMAND Subcommand [ARG] ... Subcommand arguments

Subcommands: info Show a SRPM Details list List srpm

Module Contents

Classes

<i>Srpm()</i>	Manipulates Katello engine's srpm command.
---------------	--

```
class robottelo.cli.srpm.Srpm
    Bases: robottelo.cli.base.Base
    Manipulates Katello engine's srpm command.
    command_base = srpm
    classmethod info (cls, options=None)
        Show a SRPM Info
    classmethod list (cls, options=None)
        List SRPMs
```

robottelo.cli.subnet

Usage:

```
hammer subnet [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create a subnet
delete	Delete a subnet
info	Show a subnet.
list	List of subnets
update	Update a subnet

Module Contents

Classes

<i>Subnet()</i>	Manipulates Foreman's subnets.
-----------------	--------------------------------

```
class robottelo.cli.subnet.Subnet
    Bases: robottelo.cli.base.Base
    Manipulates Foreman's subnets.
    command_base = subnet
```


robottelo.cli.subscription

Usage:

```
hammer sunscription [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

delete-manifest	Delete manifest from Red Hat provider
list	List organization subscriptions
manifest-history	obtain manifest history for subscriptions
refresh-manifest	Refresh previously imported manifest for Red Hat provider
upload	Upload a subscription manifest

Module Contents**Classes***Subscription()*

Manipulates Katello engine's subscription command.

class robottelo.cli.subscription.**Subscription**Bases: *robottelo.cli.base.Base*

Manipulates Katello engine's subscription command.

command_base = **subscription****classmethod** **upload** (*cls, options=None, timeout=None*)

Upload a subscription manifest.

classmethod **delete_manifest** (*cls, options=None, timeout=None*)

Deletes a subscription manifest.

classmethod **refresh_manifest** (*cls, options=None, timeout=None*)

Refreshes a subscription manifest.

classmethod **manifest_history** (*cls, options=None*)

Provided history for subscription manifest

robottelo.cli.syncplan

Usage:

```
hammer sync-plan [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create a sync plan
delete	Destroy a sync plan
info	Show a sync plan
list	List sync plans
update	

Module Contents

Classes

<i>SyncPlan()</i>	Manipulates Katello engine's sync-plan command.
-------------------	---

class `robottelo.cli.syncplan.SyncPlan`

Bases: `robottelo.cli.base.Base`

Manipulates Katello engine's sync-plan command.

command_base = `sync-plan`

`robottelo.cli.task`

Usage:

<code>hammer task [OPTIONS] SUBCOMMAND [ARG] ...</code>

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

list	List tasks
progress	Show the progress of the task
resume	Resume <code>all</code> tasks paused <code>in</code> error state

Module Contents

Classes

<i>Task()</i>	Manipulates Foreman's task.
---------------	-----------------------------

class `robottelo.cli.task.Task`

Bases: `robottelo.cli.base.Base`

Manipulates Foreman's task.

command_base = task

classmethod progress (*cls, options=None, return_raw_response=None*)

Shows a task progress

Usage:: hammer task progress [OPTIONS]

Options::

--id ID UUID of the task

--name NAME Name to search by

classmethod resume (*cls, options=None*)

Resumes a task

Usage: hammer task resume [OPTIONS]

Options:

--search SEARCH Resume tasks matching search string

--task-ids TASK_IDS Comma separated list of values.

--tasks TASK_NAMES Comma separated list of values.

classmethod list_tasks (*cls, options=None*)

List tasks

Usage: hammer task list [OPTIONS]

Options:

--search SEARCH List tasks matching search string

robottelo.cli.template

Usage:: hammer template [OPTIONS] SUBCOMMAND [ARG] ...

Parameters:: SUBCOMMAND Subcommand [ARG] ... Subcommand arguments

Subcommands:: add-operatingsystem Associate an operating system build-pxe-default Update the default PXE menu on all configured TFTP servers clone Clone a provision template combination Manage template combinations create Create a provisioning template delete Delete a provisioning template dump View provisioning template content info Show provisioning template details kinds List available provisioning template kinds list List provisioning templates remove-operatingsystem Disassociate an operating system update Update a provisioning template

Module Contents

Classes

Template()

Manipulates Foreman's configuration templates.

class robottelo.cli.template.**Template**

Bases: *robottelo.cli.base.Base*

Manipulates Foreman's configuration templates.

command_base = template

classmethod kinds (*cls, options=None*)
Returns list of types of templates.

classmethod add_operatingsystem (*cls, options=None*)
Adds operating system, requires “id” and “operatingsystem-id”.

classmethod remove_operatingsystem (*cls, options=None*)
Remove operating system, requires “id” and “operatingsystem-id”.

classmethod clone (*cls, options=None*)
Clone provided provisioning template

classmethod build_pxe_default (*cls, options=None*)
Build PXE default template

robottelo.cli.template_input

Usage:

```
hammer template-input [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	Subcommand
[ARG] ...	Subcommand arguments

Subcommands:

create	Create a template <code>input</code>
delete	Delete a template <code>input</code>
info	Show template <code>input</code> details
list	List template inputs

Module Contents

Classes

<code>TemplateInput()</code>	Manipulates template input.
------------------------------	-----------------------------

class robottelo.cli.template_input.**TemplateInput**

Bases: `robottelo.cli.base.Base`

Manipulates template input.

command_base = template-input

classmethod create (*cls, options=None*)

Creates a new record using the arguments passed via dictionary.

robottelo.cli.template_sync

Export:: Usage:

```
hammer export-templates [OPTIONS]
```

Options:: branch Branch in Git repo. commit-msg Custom commit message for templates export dirname The directory within Git repo containing the templates filter Export templates with names matching this regex (case-insensitive; snippets are not filtered).

location[-idl-title] Name/Title/Id of associated location location[sl-idsl-titles] REPLACE locations with given Names/Titles/Ids metadata-export-mode Specify how to handle metadata negate Negate the prefix (for purging). organization[-idl-title] Name/Title/Id of associated organization organization[sl-idsl-titles] REPLACE organizations with given Names/Titles/Ids. repo Override the default repo from settings.

Import:: Usage:

```
hammer import-templates [OPTIONS]
```

Options:: associate Associate to OS's, Locations & Organizations. Options are: always branch Branch in Git repo. dirname The directory within Git repo containing the templates filter Export templates with names matching this regex force Update templates that are locked location[-idl-title] Name/Title/Id of associated location location[sl-idsl-titles] REPLACE locations with given Names/Titles/Ids lock Lock imported templates negate Negate the prefix (for purging). organization[-idl-title] Name/Title/Id of associated organization organization[sl-idsl-titles] REPLACE organizations with given Names/Titles/Ids. prefix The string all imported templates should begin with. repo Override the default repo from settings.

Module Contents

Classes

TemplateSync()

Export/Import Satellite Templates to Git/Local Directory.

class robottelo.cli.template_sync.**TemplateSync**

Bases: *robottelo.cli.base.Base*

Export/Import Satellite Templates to Git/Local Directory.

classmethod exports (*cls, options=None*)

Export Satellite Templates to Git/Local Directory.

classmethod imports (*cls, options=None*)

Import Satellite Templates to Git/Local Directory.

robottelo.cli.user

Usage:

```
hammer user [OPTIONS] SUBCOMMAND [ARG] ...
```

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

add-role	Assign a user role
create	Create an user.
delete	Delete an user.
info	Show an user.
list	List all users.
remove-role	Remove a user role
ssh-keys	Managing User SSH Keys.
update	Update an user.

Module Contents

Classes

<i>User()</i>	Manipulates Foreman's users.
---------------	------------------------------

class robottelo.cli.user.**User**

Bases: *robottelo.cli.base.Base*

Manipulates Foreman's users.

command_base = user

classmethod **add_role** (*cls, options=None*)

Add a role to a user.

classmethod **remove_role** (*cls, options=None*)

Remove a role from user.

classmethod **ssh_keys_add** (*cls, options=None*)

Usage: hammer user ssh-keys add [OPTIONS]

Options: **-key** KEY Public SSH key **-key-file** KEY_FILE Path to a SSH public key **-location** LOCATION_NAME Location name **-location-id** LOCATION_ID **-location-title** LOCATION_TITLE Location title **-name** NAME **-organization** ORGANIZATION_NAME Organization name **-organization-id** ORGANIZATION_ID Organization ID **-organization-title** ORGANIZATION_TITLE Organization title **-user** USER_LOGIN User's login to search by **-user-id** USER_ID

classmethod **ssh_keys_delete** (*cls, options=None*)

Usage: hammer user ssh-keys delete [OPTIONS]

classmethod **ssh_keys_list** (*cls, options=None*)

Usage: hammer user ssh-keys list [OPTIONS]

classmethod **ssh_keys_info** (*cls, options=None*)

Usage: hammer user ssh-keys info [OPTIONS]

robottelo.cli.usergroup

Usage:: hammer user-group [OPTIONS] SUBCOMMAND [ARG] ...

Parameters:: SUBCOMMAND subcommand [ARG] ... subcommand arguments

Subcommands:: add-role Assign a user role add-user Associate an user add-user-group Associate an user group create Create a user group delete Delete a user group external View and manage external user groups info Show a user group list List all user groups remove-role Remove a user role remove-user Disassociate an user remove-user-group Disassociate an user group update Update a user group

Module Contents

Classes

<i>UserGroup()</i>	Manipulates Foreman's user group.
<i>UserGroupExternal()</i>	Manages Foreman external user groups.

class robottelo.cli.usergroup.**UserGroup**

Bases: *robottelo.cli.base.Base*

Manipulates Foreman's user group.

command_base = **user-group**

classmethod **add_role** (*cls, options=None*)

Assign a user role.

Usage: hammer user-group add-role [OPTIONS]

Options: **-id** ID **-name** NAME Name to search by **-role** ROLE_NAME User role name **-role-id** ROLE_ID

classmethod **add_user** (*cls, options=None*)

Associate an user.

Usage: hammer user-group add-user [OPTIONS]

Options: **-id** ID **-name** NAME Name to search by **-user** USER_LOGIN User's login to search by **-user-id** USER_ID

classmethod **add_user_group** (*cls, options=None*)

Associate an user group.

Usage: hammer user-group add-user-group [OPTIONS]

Options: **-id** ID **-name** NAME Name to search by **-user-group** USER_GROUP_NAME Name to search by **-user-group-id** USER_GROUP_ID

classmethod **remove_role** (*cls, options=None*)

Remove a user role.

Usage: hammer user-group remove-role [OPTIONS]

Options: **-id** ID **-name** NAME Name to search by **-role** ROLE_NAME User role name **-role-id** ROLE_ID

classmethod **remove_user** (*cls, options=None*)

Disassociate an user.

Usage: hammer user-group remove-user [OPTIONS]

Options: **-id** ID **-name** NAME Name to search by **-user** USER_LOGIN User's login to search by **-user-id** USER_ID

classmethod **remove_user_group** (*cls, options=None*)

Disassociate an user group.

Usage: hammer user-group remove-user-group [OPTIONS]

Options: **-id** ID **-name** NAME Name to search by **-user-group** USER_GROUP_NAME Name to search by **-user-group-id** USER_GROUP_ID

class robottelo.cli.usergroup.**UserGroupExternal**

Bases: *robottelo.cli.base.Base*

Manages Foreman external user groups.

Usage: hammer user-group external [OPTIONS] SUBCOMMAND [ARG] ...

Subcommands: create Create an external user group linked to a user group delete Delete an external user group
info Show an external user group for user group list List all external user groups for user group refresh
Refresh external user group update Update external user group

command_base = **user-group external**

classmethod **refresh** (*cls, options=None*)

classmethod **create** (*cls, options=None*)

Create external user group

robottelo.cli.virt_who_config

Usage: hammer virt-who-config [OPTIONS] SUBCOMMAND [ARG] ...

Parameters:

SUBCOMMAND	subcommand
[ARG] ...	subcommand arguments

Subcommands:

create	Create a virt-who configuration
delete	Delete a virt-who configuration
deploy	Download and execute script for the specified virt-who configuration
fetch	Renders a deploy script for the specified virt-who configuration
info	Show a virt-who configuration
list	List of virt-who configurations
update	Update a virt-who configuration

Module Contents

Classes

VirtWhoConfig()

Manipulates virt-who configuration.

class robottelo.cli.virt_who_config.**VirtWhoConfig**

Bases: *robottelo.cli.base.Base*

Manipulates virt-who configuration.

command_base = **virt-who-config**

classmethod **fetch** (*cls, options=None, output_format=None*)

Renders a deploy script for the specified virt-who configuration

classmethod **deploy** (*cls, options=None*)

runs hammer virt-who-config deploy -id <x> which runs the script on the satellite server

Parameters `options` – `id` required

Returns Results of the command

`robottelo.config`

Submodules

`robottelo.config.base`

Define and instantiate the configuration class for Robottelo.

Module Contents

Classes

<code>INIReader(path)</code>	ConfigParser wrapper able to cast value when reading INI options.
<code>FeatureSettings()</code>	Settings related to a feature.
<code>ServerSettings(*args, **kwargs)</code>	Satellite server settings definitions.
<code>BugzillaSettings(*args, **kwargs)</code>	Bugzilla server settings definitions.
<code>CapsuleSettings(*args, **kwargs)</code>	Clients settings definitions.
<code>CertsSettings(*args, **kwargs)</code>	Katello-certs settings definitions.
<code>ClientsSettings(*args, **kwargs)</code>	Clients settings definitions.
<code>ContainerRepositorySettings(*args, **kwargs)</code>	Settings for syncing containers from container registries
<code>DistroSettings(*args, **kwargs)</code>	Distro settings definitions.
<code>DockerSettings(*args, **kwargs)</code>	Docker settings definitions.
<code>AzureRMSettings(*args, **kwargs)</code>	Azure Resource Manager settings definitions.
<code>EC2Settings(*args, **kwargs)</code>	AWS EC2 settings definitions.
<code>FakeManifestSettings(*args, **kwargs)</code>	Fake manifest settings definitions.
<code>GCESettings(*args, **kwargs)</code>	Google Compute Engine settings definitions.
<code>RHSSOSettings(*args, **kwargs)</code>	RHSSO settings definitions.
<code>LDAPSettings(*args, **kwargs)</code>	LDAP settings definitions.
<code>LDAPIPASettings(*args, **kwargs)</code>	LDAP freeIPA settings definitions.
<code>LibvirtHostSettings(*args, **kwargs)</code>	Libvirt host settings definitions.
<code>FakeCapsuleSettings(*args, **kwargs)</code>	Fake Capsule settings definitions.
<code>RHEVSettings(*args, **kwargs)</code>	RHEV settings definitions.
<code>VmWareSettings(*args, **kwargs)</code>	VmWare settings definitions.
<code>DiscoveryISOSettings(*args, **kwargs)</code>	Discovery ISO name settings definition.
<code>OscapSettings(*args, **kwargs)</code>	Oscap settings definitions.
<code>OSPSettings(*args, **kwargs)</code>	OSP settings definitions.
<code>OstreeSettings(*args, **kwargs)</code>	Ostree settings definitions.
<code>PerformanceSettings(*args, **kwargs)</code>	Performance settings definitions.
<code>RHAISettings(*args, **kwargs)</code>	RHAI settings definitions.
<code>SSHClientSettings(*args, **kwargs)</code>	SSHClient settings definitions.
<code>VlanNetworkSettings(*args, **kwargs)</code>	Vlan Network settings definitions.
<code>UpgradeSettings(*args, **kwargs)</code>	Satellite upgrade settings definitions.
<code>SharedFunctionSettings(*args, **kwargs)</code>	Shared function settings definitions.
<code>VirtWhoSettings(*args, **kwargs)</code>	VirtWho settings definitions.

Continued on next page

Table 73 – continued from previous page

<code>ReportPortalSettings(*args, **kwargs)</code>	Report portal settings definitions.
<code>Settings()</code>	Robottelo's settings representation.
<code>HttpProxySettings(*args, **kwargs)</code>	Http Proxy settings definitions.

Functions

<code>get_project_root()</code>	Return the path to the Robottelo project root directory.
---------------------------------	--

`robottelo.config.base.LOGGER`

`robottelo.config.base.SETTINGS_FILE_NAME = robottelo.properties`

exception `robottelo.config.base.ImproperlyConfigured`

Bases: Exception

Indicates that Robottelo somehow is improperly configured.

For example, if settings file can not be found or some required configuration is not defined.

`robottelo.config.base.get_project_root()`

Return the path to the Robottelo project root directory.

Returns A directory path.

Return type str

class `robottelo.config.base.INIReader` (*path*)

Bases: object

ConfigParser wrapper able to cast value when reading INI options.

cast_boolean

cast_dict

cast_list

cast_logging_level

cast_tuple

cast_webdriver_desired_capabilities

get (*self, section, option, default=None, cast=None*)

Read an option from a section of a INI file.

First try to lookup for the value as an environment variable having the following format: ROBOT-TELO_{SECTION}_{OPTION}.

The default value will return if the look up option is not available. The value will be cast using a callable if specified otherwise a string will be returned.

Parameters

- **section** – Section to look for.
- **option** – Option to look for.
- **default** – The value that should be used if the option is not defined.
- **cast** – If provided the value will be cast using the cast provided.

has_section (*self*, *section*)
Check if section is available.

class robottelo.config.base.**FeatureSettings**

Bases: object

Settings related to a feature.

Create a instance of this class and assign attributes to map to the feature options.

read (*self*, *reader*)
Subclasses must implement this method in order to populate itself with expected settings values.

Parameters **reader** – An INIReader instance to read the settings.

validate (*self*)
Subclasses must implement this method in order to validate the settings and raise `ImproperlyConfigured` if any issue is found.

class robottelo.config.base.**ServerSettings** (*args, **kwargs)

Bases: *robottelo.config.base.FeatureSettings*

Satellite server settings definitions.

version

read (*self*, *reader*)
Read and validate Satellite server settings.

validate (*self*)
Subclasses must implement this method in order to validate the settings and raise `ImproperlyConfigured` if any issue is found.

get_credentials (*self*)
Return credentials for interacting with a Foreman deployment API.

Returns A username-password pair.

Return type tuple

get_url (*self*)
Return the base URL of the Foreman deployment being tested.

The following values from the config file are used to build the URL:

- [server] scheme (default: https)
- [server] hostname (required)
- [server] port (default: none)

Setting port to 80 does *not* imply that scheme is 'https'. If port is 80 and scheme is unset, scheme will still default to 'https'.

Returns A URL.

Return type str

get_pub_url (*self*)
Return the pub URL of the server being tested.

The following values from the config file are used to build the URL:

- main.server.hostname (required)

Returns The pub directory URL.

Return type str

get_cert_rpm_url (*self*)

Return the Katello cert RPM URL of the server being tested.

The following values from the config file are used to build the URL:

- `main.server.hostname` (required)

Returns The Katello cert RPM URL.

Return type str

class `robottelo.config.base.BugzillaSettings` (*args, **kwargs)

Bases: `robottelo.config.base.FeatureSettings`

Bugzilla server settings definitions.

read (*self*, *reader*)

Read and validate Bugzilla server settings.

validate (*self*)

This section is lazily validated on `.issue_handlers.bugzilla`.

class `robottelo.config.base.CapsuleSettings` (*args, **kwargs)

Bases: `robottelo.config.base.FeatureSettings`

Clients settings definitions.

hostname

read (*self*, *reader*)

Read clients settings.

validate (*self*)

Validate capsule settings.

class `robottelo.config.base.CertsSettings` (*args, **kwargs)

Bases: `robottelo.config.base.FeatureSettings`

Katello-certs settings definitions.

read (*self*, *reader*)

Read certs settings.

validate (*self*)

Validate certs settings.

class `robottelo.config.base.ClientsSettings` (*args, **kwargs)

Bases: `robottelo.config.base.FeatureSettings`

Clients settings definitions.

read (*self*, *reader*)

Read clients settings.

validate (*self*)

Validate clients settings.

class `robottelo.config.base.ContainerRepositorySettings` (*args, **kwargs)

Bases: `robottelo.config.base.FeatureSettings`

Settings for syncing containers from container registries

section = container_repo

```

repo_config_required = ['label', 'registry_url', 'registry_username', 'registry_passwo
read (self, reader)
    Read container repo settings and associated yaml file
validate (self)
    Subclasses must implement this method in order to validate the settings and raise
    ImproperlyConfigured if any issue is found.
_validate_registry_configs (self, configs)
class robottelo.config.base.DistroSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings
    Distro settings definitions.
    read (self, reader)
        Read distro settings.
    validate (self)
        Validate distro settings.
class robottelo.config.base.DockerSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings
    Docker settings definitions.
    read (self, reader)
        Read docker settings.
    validate (self)
        Validate docker settings.
class robottelo.config.base.AzureRMSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings
    Azure Resource Manager settings definitions.
    read (self, reader)
        Read AzureRM settings.
    validate (self)
        Validate AzureRM settings.
class robottelo.config.base.EC2Settings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings
    AWS EC2 settings definitions.
    read (self, reader)
        Read AWS EC2 settings.
    validate (self)
        Validate AWS EC2 settings.
class robottelo.config.base.FakeManifestSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings
    Fake manifest settings definitions.
    read (self, reader)
        Read fake manifest settings.
    validate (self)
        Validate fake manifest settings.

```

```
class robottelo.config.base.GCESettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings
```

Google Compute Engine settings definitions.

```
read (self, reader)
    Read GCE settings.
```

```
validate (self)
    Validate GCE settings.
```

```
class robottelo.config.base.RHSSOSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings
```

RHSSO settings definitions.

```
read (self, reader)
    Read LDAP settings.
```

```
validate (self)
    Validate RHSSO settings.
```

```
class robottelo.config.base.LDAPSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings
```

LDAP settings definitions.

```
read (self, reader)
    Read LDAP settings.
```

```
validate (self)
    Validate LDAP settings.
```

```
class robottelo.config.base.LDAPIPASettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings
```

LDAP freeIPA settings definitions.

```
read (self, reader)
    Read LDAP freeIPA settings.
```

```
validate (self)
    Validate LDAP freeIPA settings.
```

```
class robottelo.config.base.LibvirtHostSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings
```

Libvirt host settings definitions.

```
read (self, reader)
    Read libvirt host settings.
```

```
validate (self)
    Validate libvirt host settings.
```

```
class robottelo.config.base.FakeCapsuleSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings
```

Fake Capsule settings definitions.

```
read (self, reader)
    Read fake capsule settings
```

```
validate (self)
    Validate fake capsule settings.
```

```
class robottelo.config.base.RHEVSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings

    RHEV settings definitions.

    read (self, reader)
        Read rhev settings.

    validate (self)
        Validate rhev settings.

class robottelo.config.base.VmWareSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings

    VmWare settings definitions.

    read (self, reader)
        Read vmware settings.

    validate (self)
        Validate vmware settings.

class robottelo.config.base.DiscoveryISOSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings

    Discovery ISO name settings definition.

    read (self, reader)
        Read discovery iso setting.

    validate (self)
        Validate discovery iso name setting.

class robottelo.config.base.OscapSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings

    Oscap settings definitions.

    read (self, reader)
        Read Oscap settings.

    validate (self)
        Validate Oscap settings.

class robottelo.config.base.OSPSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings

    OSP settings definitions.

    read (self, reader)
        Read osp settings.

    validate (self)
        Validate osp settings.

class robottelo.config.base.OstreeSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings

    Ostree settings definitions.

    read (self, reader)
        Read Ostree settings.

    validate (self)
        Validate Ostree settings.
```

```
class robottelo.config.base.PerformanceSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings

    Performance settings definitions.

    read (self, reader)
        Read performance settings.

    validate (self)
        Validate performance settings.

class robottelo.config.base.RHAISettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings

    RHAISettings definitions.

    read (self, reader)
        Read RHAISettings.

    validate (self)
        Validate RHAISettings.

class robottelo.config.base.SSHClientSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings

    SSHClient settings definitions.

    command_timeout

    connection_timeout

    read (self, reader)
        Read SSHClient settings.

    validate (self)
        Validate SSHClient settings.

class robottelo.config.base.VlanNetworkSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings

    Vlan Network settings definitions.

    read (self, reader)
        Read Vlan Network settings.

    validate (self)
        Validate Vlan Network settings.

class robottelo.config.base.UpgradeSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings

    Satellite upgrade settings definitions.

    read (self, reader)
        Read and validate Satellite server settings.

    validate (self)
        Subclasses must implement this method in order to validate the settings and raise
        ImproperlyConfigured if any issue is found.

class robottelo.config.base.SharedFunctionSettings (*args, **kwargs)
    Bases: robottelo.config.base.FeatureSettings

    Shared function settings definitions.

    MAX_SHARE_TIMEOUT = 86400
```


read (*self*, *reader*)
Read shared settings.

validate (*self*)
Validate the shared settings

class robottelo.config.base.VirtWhoSettings (*args, **kwargs)

Bases: *robottelo.config.base.FeatureSettings*

VirtWho settings definitions.

read (*self*, *reader*)
Read virtwho settings.

validate (*self*)
Validate virtwho settings.

class robottelo.config.base.ReportPortalSettings (*args, **kwargs)

Bases: *robottelo.config.base.FeatureSettings*

Report portal settings definitions.

read (*self*, *reader*)
Read Report portal settings.

validate (*self*)
Validate Report portal settings.

class robottelo.config.base.Settings

Bases: object

Robottelo's settings representation.

configured
Returns True if the settings have already been configured.

all_features
List all expected feature settings sections.

configure (*self*, *settings_path=None*)
Read the settings file and parse the configuration.

Parameters *settings_path* (*str*) – path to settings file to read. If None, looks in the project root for a file named 'robottelo.properties'.

Raises ImproperlyConfigured if any issue is found during the parsing or validation of the configuration.

_read_robottelo_settings (*self*)
Read Robottelo's general settings.

_validate_robottelo_settings (*self*)
Validate Robottelo's general settings.

_configure_entities (*self*)
Configure NailGun's entity classes.

Do the following:

- Set `entity_mixins.CREATE_MISSING` to True. This causes method `EntityCreateMixin.create_raw` to generate values for empty and required fields.
- Set `nailgun.entity_mixins.DEFAULT_SERVER_CONFIG` to whatever is returned by `robottelo.helpers.get_nailgun_config()`. See `robottelo.entity_mixins.Entity` for more information on the effects of this.

- Set a default value for `nailgun.entities.GPGKey.content`.

`_configure_airgun` (*self*)
 Pass required settings to AirGun

`_configure_logging` (*self*)
 Configure logging for the entire framework.

If a config named `logging.conf` exists in Robottelo's root directory, the logger is configured using the options in that file. Otherwise, a custom logging output format is set, and default values are used for all other logging options.

`_configure_third_party_logging` (*self*)
 Increase the level of third party packages logging.

class `robottelo.config.base.HttpProxySettings` (*args, **kwargs)

Bases: `robottelo.config.base.FeatureSettings`

Http Proxy settings definitions.

read (*self*, reader)
 Read Http Proxy settings.

validate (*self*)
 Validate Http Proxy settings.

`robottelo.config.casts`

Configuration casts to help typing the settings.

Module Contents

Classes

<code>Boolean()</code>	Cast a string to boolean.
<code>List()</code>	Cast a comma separated string to a list.
<code>LoggingLevel()</code>	Cast a string to a logging level.
<code>Tuple()</code>	Cast a comma separated string to a tuple.
<code>Dict()</code>	Cast a comma separated list of key=value to a dict.
<code>WebdriverDesiredCapabilities()</code>	Cast a comma separated list of key=value to a

class `robottelo.config.casts.Boolean`

Bases: `object`

Cast a string to boolean.

String values 1, yes, true, on will result in python's True. String values 0, no, false, off will result in python's False.

Parameters `value` (*str*) – A string to cast to boolean.

`_booleans`

`__call__` (*self*, value)

class `robottelo.config.casts.List`

Bases: `object`

Cast a comma separated string to a list.

Parameters **value** (*str*) – A comma separated string to cast to a list.

`__call__` (*self*, *value*)

class `robottelo.config.castst.LoggingLevel`

Bases: `object`

Cast a string to a logging level.

Parameters **value** (*str*) – A string to cast to a logging level.

`_logging_levels`

`__call__` (*self*, *value*)

class `robottelo.config.castst.Tuple`

Bases: `robottelo.config.castst.List`

Cast a comma separated string to a tuple.

Parameters **value** (*str*) – A comma separated string to cast to a tuple.

`__call__` (*self*, *value*)

class `robottelo.config.castst.Dict`

Bases: `robottelo.config.castst.List`

Cast a comma separated list of key=value to a dict.

Parameters **value** (*str*) – A comma separated string to cast to a dict.

`__call__` (*self*, *value*)

class `robottelo.config.castst.WebdriverDesiredCapabilities`

Bases: `robottelo.config.castst.Dict`

Cast a comma separated list of key=value to a webdriver.DesiredCapabilities dict.

Convert values `true` and `false` (ignore case) to a proper boolean.

Parameters **value** (*str*) – A comma separated string to cast to a webdriver.DesiredCapabilities dict.

`__call__` (*self*, *value*)

Package Contents

Classes

`Settings()`

Robottelo's settings representation.

class `robottelo.config.Settings`

Bases: `object`

Robottelo's settings representation.

configured

Returns True if the settings have already been configured.

all_features

List all expected feature settings sections.

configure (*self*, *settings_path=None*)

Read the settings file and parse the configuration.

Parameters **settings_path** (*str*) – path to settings file to read. If None, looks in the project root for a file named ‘robottelo.properties’.

Raises ImproperlyConfigured if any issue is found during the parsing or validation of the configuration.

_read_robottelo_settings (*self*)

Read Robottelo’s general settings.

_validate_robottelo_settings (*self*)

Validate Robottelo’s general settings.

_configure_entities (*self*)

Configure NailGun’s entity classes.

Do the following:

- **Set `entity_mixins.CREATE_MISSING` to `True`. This causes method `EntityCreateMixin.create_raw` to generate values for empty and required fields.**
- **Set `nailgun.entity_mixins.DEFAULT_SERVER_CONFIG` to whatever is returned by `robottelo.helpers.get_nailgun_config()`. See `robottelo.entity_mixins.Entity` for more information on the effects of this.**
- Set a default value for `nailgun.entities.GPGKey.content`.

_configure_airgun (*self*)

Pass required settings to AirGun

_configure_logging (*self*)

Configure logging for the entire framework.

If a config named `logging.conf` exists in Robottelo’s root directory, the logger is configured using the options in that file. Otherwise, a custom logging output format is set, and default values are used for all other logging options.

_configure_third_party_logging (*self*)

Increase the level of third party packages logging.

`robottelo.config.settings`

`robottelo.decorators`

Implements various decorators

Subpackages

`robottelo.decorators.func_shared`

Submodules

`robottelo.decorators.func_shared.base`

Module Contents

Classes

BaseStorageHandler()

class robottelo.decorators.func_shared.base.**BaseStorageHandler**

Bases: object

static encode (*data*)

static decode (*data*)

lock (*self, lock_key*)

Return the storage locker context manager

when_lock_acquired (*self, data*)

called when the lock is acquired to do some added action

get (*self, key*)

Return the key value

set (*self, key, value*)

Write the value of key to storage

robottelo.decorators.func_shared.file_storage

Module Contents

Classes

FileStorageHandler(*root_dir=None, create=True, Key value file storage handler.*
lock_timeout=LOCK_TIMEOUT)

Functions

get_temp_dir()

_get_root_dir(*create=True*)

robottelo.decorators.func_shared.file_storage.**TEMP_ROOT_DIR** = robottelo

robottelo.decorators.func_shared.file_storage.**TEMP_FUNC_SHARED_DIR** = shared_functions

robottelo.decorators.func_shared.file_storage.**SHARED_DIR**

robottelo.decorators.func_shared.file_storage.**logger**

robottelo.decorators.func_shared.file_storage.**LOCK_TIMEOUT** = 7200

robottelo.decorators.func_shared.file_storage.**get_temp_dir**()

robottelo.decorators.func_shared.file_storage.**_get_root_dir** (*create=True*)

class robottelo.decorators.func_shared.file_storage.**FileStorageHandler** (*root_dir=None,*
create=True,
lock_timeout=LOCK_TIMEOUT)

Bases: *robottelo.decorators.func_shared.base.BaseStorageHandler*

Key value file storage handler.

root_dir

get_key_file_path (*self, key*)

lock (*self, key*)

Return the storage locker context manager

when_lock_acquired (*self, handler*)

Write the process id to file handler

get (*self, key*)

Return the key value :type key: str

set (*self, key, value*)

Write the value of key

robottelo.decorators.func_shared.redis_storage

Module Contents

Classes

RedisStorageHandler(host=REDIS_HOST, Redis Key value storage handler
port=REDIS_PORT, db=REDIS_DB,
password=REDIS_PASSWORD,
lock_timeout=LOCK_TIMEOUT)

robottelo.decorators.func_shared.redis_storage.**redis**

robottelo.decorators.func_shared.redis_storage.**REDIS_HOST** = localhost

robottelo.decorators.func_shared.redis_storage.**REDIS_PORT** = 6379

robottelo.decorators.func_shared.redis_storage.**REDIS_DB** = 0

robottelo.decorators.func_shared.redis_storage.**REDIS_PASSWORD**

robottelo.decorators.func_shared.redis_storage.**LOCK_TIMEOUT** = 7200

class robottelo.decorators.func_shared.redis_storage.**RedisStorageHandler** (*host=REDIS_HOST,*
port=REDIS_PORT,
db=REDIS_DB,
pass-
word=REDIS_PASSWORD,
lock_timeout=LOCK_TIMEOUT)

Bases: *robottelo.decorators.func_shared.base.BaseStorageHandler*

Redis Key value storage handler

client

lock (*self, key, timeout=None*)

Return the storage locker context manager

when_lock_acquired (*self, lock_object*)
called when the lock is acquired to do some added action

get (*self, key*)
Return the key value

set (*self, key, value*)
Write the value of key

robottelo.decorators.func_shared.shared

Shared function is a decorator, that enable a function once called to store the results to storage, any ulterior call from the same or other processes will return the stored results, which make the shared function results persistent.

Note: Shared function store it's data as json. The results of the decorated function must be json compatible.

Usage:

```

from robottelo.decorators.func_shared.shared import shared

@shared
def module_level_shared(*args, **kwargs):
    # do some
    # the result of a shared function must be json compatible
    return any_data

class SomeTestCase1(TestCase):

    @shared
    def _shared_function(cls):

        org = make_org()
        # upload manifest
        repo = make_repository()
        return dict(org=org, repo=repo)

    @classmethod
    @shared
    def setUpClass(cls):

        data = cls._shared_function()
        other_data = module_level_shared()

        cls.org = data['org']
        cls.repo = data['repo']
        return

# the shared function can be called an other time to be able to initiate
# specific data

class SomeTestCase2(TestCase):

    @classmethod
    @shared(inject=True, injected_kw='_injected')

```

(continues on next page)

(continued from previous page)

```

def setUpClass(cls, org=None, repo=None, _injected=False):

    if _injected:
        cls.org = org
        cls.repo = repo
    else:
        # create the org
        cls.org = make_org()
        # upload manifest
        cls.repo = make_repository()

    # create a virtual machine

    # shared function with injected=True, must return a dict
    # the resulting dictionary will be injected in other calls as
    # kwargs, an added bool kw argument by default named _injected
    # should be added to the function kwargs, to be able to be notified
    # that the kwargs are injected from already stored result
    return dict(org=cls.org, repo=cls.repo)

# in case we do not want the injected key word in kwargs
# simply , declare injected_kw=None
@classmethod
@shared(inject=True, injected_kw=None)
def shared_class_method(cls, org=None, repo=None):
    if org is not None:
        cls.org = org
    else:
        # create the org
        cls.org = make_org()
        # upload manifest
    if repo_id is not None:
        cls.repo = repo
    else:
        cls.repo = make_repository()

    # create a virtual machine

    return dict(org=cls.org, repo=cls.repo)

```

Module Contents

Classes

<code>__SharedFunction</code> (function_key, function, args=None, kwargs=None, retries=DEFAULT_CALL_RETRIES, storage_handler=None, timeout=SHARE_DEFAULT_TIMEOUT, inject=False, injected_kw='_inject')	Internal class helper that is created each time the shared function is
--	--

Functions

<code>__set_configured(value)</code>	
<code>__check_config()</code>	
<code>enable_shared_function(value)</code>	force and override settings, by setting the global use shared data
<code>set_default_scope(value)</code>	Set the default namespace scope
<code>__get_default_scope()</code>	Return the shared function default scope
<code>__get_default_storage_handler()</code>	Return the storage handler instance
<code>__get_kwargs_md5(**kwargs)</code>	Create an md5 hexdigest from kwargs
<code>__get_scope_name(scope=None, scope_kwargs=None, scope_context=None)</code>	
<code>__get_function_name(function, class_name=None, kwargs=None)</code>	Return a string representation of the function as
<code>__get_function_name_key(function_name, scope=None, scope_kwargs=None, scope_context=None)</code>	
<code>shared(function_=None, scope=__get_default_scope, scope_context=None, scope_kwargs=None, timeout=SHARE_DEFAULT_TIMEOUT, retries=DEFAULT_CALL_RETRIES, function_kw=None, inject=False, injected_kw='injected')</code>	Generic function sharing, share the results of any decorated function.

```

robottelo.decorators.func_shared.shared.logger
robottelo.decorators.func_shared.shared._storage_handlers
robottelo.decorators.func_shared.shared.DEFAULT_STORAGE_HANDLER = file
robottelo.decorators.func_shared.shared.ENABLED = False
robottelo.decorators.func_shared.shared.NAMESPACE_SCOPE
robottelo.decorators.func_shared.shared.SHARE_DEFAULT_TIMEOUT = 86400
robottelo.decorators.func_shared.shared.DEFAULT_CALL_RETRIES = 2
robottelo.decorators.func_shared.shared.__configured = False
robottelo.decorators.func_shared.shared._NAMESPACE_SCOPE_KEY_TYPE = shared_function
robottelo.decorators.func_shared.shared._DEFAULT_CLASS_NAME_DEPTH = 3
robottelo.decorators.func_shared.shared._STATE_READY = READY
robottelo.decorators.func_shared.shared._STATE_FAILED = FAILED
robottelo.decorators.func_shared.shared._DATETIME_FORMAT = %Y-%m-%dT%H:%M:%S
robottelo.decorators.func_shared.shared._SERVER_CERT_MD5
robottelo.decorators.func_shared.shared.__set_configured(value)
robottelo.decorators.func_shared.shared.__check_config()
robottelo.decorators.func_shared.shared.enable_shared_function(value)
    force and override settings, by setting the global use shared data attribute
robottelo.decorators.func_shared.shared.set_default_scope(value)
    Set the default namespace scope :type value: str or callable
robottelo.decorators.func_shared.shared.__get_default_scope()
    Return the shared function default scope

```

`robottelo.decorators.func_shared.shared._get_default_storage_handler()`
Return the storage handler instance

exception `robottelo.decorators.func_shared.shared.SharedFunctionError`
Bases: Exception
Shared function related exception

exception `robottelo.decorators.func_shared.shared.SharedFunctionException`
Bases: Exception
Shared function call exception when not able to restore the original exception

class `robottelo.decorators.func_shared.shared._SharedFunction` (*function_key*,
function,
args=None,
kwargs=None, *re-*
tries=DEFAULT_CALL_RETRIES,
stor-
age_handler=None,
time-
out=SHARE_DEFAULT_TIMEOUT,
inject=False, *in-*
jected_kw='_inject')

Bases: object

Internal class helper that is created each time the shared function is launched and group all the necessary functionality

storage

key

transaction

`_encode_result_kwargs` (*self*, *kwargs*)
look for some special kwargs and convert them

`_call_function` (*self*)

`_has_result_expired` (*self*, *creation_datetime*)

`__call__` (*self*)

`robottelo.decorators.func_shared.shared._get_kwargs_md5` (***kwargs*)
Create an md5 hexdigest from kwargs

`robottelo.decorators.func_shared.shared._get_scope_name` (*scope=None*,
scope_kwargs=None,
scope_context=None)

`robottelo.decorators.func_shared.shared._get_function_name` (*function*,
class_name=None,
kwargs=None)
Return a string representation of the function as `module_path.Class_name.function_name`

note: the class name is the first parent class

`robottelo.decorators.func_shared.shared._get_function_name_key` (*function_name*,
scope=None,
scope_kwargs=None,
scope_context=None)

```
robottelo.decorators.func_shared.shared.shared(function_=None,
                                                scope=_get_default_scope,
                                                scope_context=None,
                                                scope_kwargs=None,          time-
                                                out=SHARE_DEFAULT_TIMEOUT,
                                                retries=DEFAULT_CALL_RETRIES,
                                                function_kw=None,          inject=False,
                                                injected_kw='_injected')
```

Generic function sharing, share the results of any decorated function. Any parallel pytest xdist worker will wait for this function to finish

Parameters

- **function** (*callable*) – the function that is intended to be shared
- **scope** (*str or callable*) – this parameter will define the namespace of data sharing
- **scope_context** (*str*) – an added context string if applicable, of a concrete sharing in combination with scope and function.
- **scope_kwargs** (*dict*) – kwargs to be passed to scope if it is a callable
- **timeout** (*int*) – the time in seconds to wait for waiting the shared function
- **retries** (*int*) – if the shared function call fail, how much time should retry before setting the call with in failure state
- **function_kw** (*list*) – The function kwargs to use as an additional scope, an md5 hexdigest of that kwargs will be created and added to the storage scope, that way we should have different stored values for different kw values.
- **inject** (*bool*) – whether to recall the function with injecting the result as ****kwargs**
- **injected_kw** (*str*) – the kw arg to set to True to inform the function that the kwargs was injected from a saved storage

Package Contents

exception `robottelo.decorators.func_shared.SharedFunctionError`

Bases: `Exception`

Shared function related exception

exception `robottelo.decorators.func_shared.SharedFunctionException`

Bases: `Exception`

Shared function call exception when not able to restore the original exception

Submodules

`robottelo.decorators.func_locker`

Implements test function locking, using `pytest_services` file locking

Usage:

```

from robottelo.decorators.func_locker import (
    locking_function,
    lock_function,
)

# in many cases we have tests that need some test functions to run isolated
# from other py.test workers when used in --boxed mode
class SomeTestCase(TestCase):

    @classmethod
    @lock_function
    def setUpClass(cls):
        pass

# in other cases we want only a portion of the test function to be isolated
class SomeTestCase(TestCase):

    @classmethod
    def setUpClass(cls):

        with locking_function(cls.setUpClass,
                              scope_context='publish_puppet_class'):
            # call the publish function

# some tests can be in conflicts with other tests parts
class SomeTestCase(TestCase):

    @lock_function
    def test_to_lock(self):
        pass

    def test_that_conflict_with_test_to_lock(self):
        with locking_function(self.test_to_lock):
            # do some operations that conflict with test_to_lock

```

Module Contents

Functions

<code>set_default_scope(value)</code>	Set the default namespace scope
<code>_get_default_scope()</code>	
<code>get_temp_dir()</code>	
<code>_get_temp_lock_function_dir(create=True)</code>	
<code>_get_scope_path(scope, scope_kwarg=None, scope_context=None, create=True)</code>	Returns the scopes path and create it if create is true
<code>_get_function_name(function, class_name=None)</code>	Return a string representation of the function as
<code>_get_function_name_lock_path(function_name, scope=None, scope_kwarg=None, scope_context=None)</code>	Return the path of the file to lock
<code>_check_deadlock(lock_file_path, process_id)</code>	To prevent process deadlock, raise exception if the file content is the
<code>_write_content(handler, content)</code>	write content to locked file

Continued on next page

Table 83 – continued from previous page

<code>lock_function</code> (function=None, scope=_get_default_scope, scope_kwargs=None, out=LOCK_DEFAULT_TIMEOUT)	scope_context=None, time-	Generic function locker, lock any decorated function. Any parallel
<code>locking_function</code> (function, scope=_get_default_scope, scope_kwargs=None, out=LOCK_DEFAULT_TIMEOUT)	scope_context=None, time-	Lock a function in combination with a scope and scope_context.

`robottelo.decorators.func_locker.logger`

`robottelo.decorators.func_locker.TEMP_ROOT_DIR = robottelo`

`robottelo.decorators.func_locker.TEMP_FUNC_LOCK_DIR = lock_functions`

`robottelo.decorators.func_locker.LOCK_DIR`

`robottelo.decorators.func_locker.LOCK_DEFAULT_TIMEOUT = 1800`

`robottelo.decorators.func_locker.LOCK_FILE_NAME_EXT = lock`

`robottelo.decorators.func_locker.LOCK_DEFAULT_SCOPE`

`robottelo.decorators.func_locker._DEFAULT_CLASS_NAME_DEPTH = 3`

exception `robottelo.decorators.func_locker.FunctionLockerError`

Bases: Exception

the default function locker error

`robottelo.decorators.func_locker.set_default_scope` (*value*)

Set the default namespace scope

`robottelo.decorators.func_locker._get_default_scope` ()

`robottelo.decorators.func_locker.get_temp_dir` ()

`robottelo.decorators.func_locker._get_temp_lock_function_dir` (*create=True*)

`robottelo.decorators.func_locker._get_scope_path` (*scope*, *scope_kwargs=None*, *scope_context=None*, *create=True*)

Returns the scopes path and create it if create is true

`robottelo.decorators.func_locker._get_function_name` (*function*, *class_name=None*)

Return a string representation of the function as module_path.Class_name.function_name

note: the class name is the first parent class

`robottelo.decorators.func_locker._get_function_name_lock_path` (*function_name*, *scope=None*, *scope_kwargs=None*, *scope_context=None*)

Return the path of the file to lock

`robottelo.decorators.func_locker._check_deadlock` (*lock_file_path*, *process_id*)

To prevent process deadlock, raise exception if the file content is the same as process_id

note: this function is called before the lock

`robottelo.decorators.func_locker._write_content` (*handler, content*)
write content to locked file

`robottelo.decorators.func_locker.lock_function` (*function=None, scope=_get_default_scope, scope_context=None, scope_kwargs=None, timeout=LOCK_DEFAULT_TIMEOUT*)

Generic function locker, lock any decorated function. Any parallel pytest xdist worker will wait for this function to finish

Parameters

- **function** (*callable*) – the function that is intended to be locked
- **scope** (*str or callable*) – this parameter will define the namespace of locking
- **scope_context** (*str*) – an added context string if applicable, of a concrete lock in combination with scope and function.
- **scope_kwargs** (*dict*) – kwargs to be passed to scope if is a callable
- **timeout** (*int*) – the time in seconds to wait for acquiring the lock

`robottelo.decorators.func_locker.locking_function` (*function, scope=_get_default_scope, scope_context=None, scope_kwargs=None, timeout=LOCK_DEFAULT_TIMEOUT*)

Lock a function in combination with a scope and scope_context. Any parallel pytest xdist worker will wait for this function to finish.

Parameters

- **function** (*callable*) – the function that is intended to be locked
- **scope** (*str or callable*) – this parameter will define the namespace of locking
- **scope_context** (*str*) – an added context string if applicable, of a concrete lock in combination with scope and function.
- **scope_kwargs** (*dict*) – kwargs to be passed to scope if is a callable
- **timeout** (*int*) – the time in seconds to wait for acquiring the lock

`robottelo.decorators.host`

Implements decorator regarding satellite host

Module Contents

Functions

<code>skip_if_os(*versions)</code>	Decorator to skip tests based on host version
------------------------------------	---

`robottelo.decorators.host.LOGGER`

`robottelo.decorators.host.skip_if_os(*versions)`

Decorator to skip tests based on host version

If the calling function uses 'RHEL6' - test will be skipped for RHEL6, but will run for whatever another version, e.g, RHEL5, RHEL6.1, RHEL7, and so on

Note: If the version can't be obtained, tests will run

Usage:

To skip a specific test:

```
from robottelo.decorators.host import skip_if_host_is

@skip_if_os('RHEL6')
def test_hostgroup_create():
    # test code continues here
```

Parameters `versions` (*tuple*) – args containing host versions for which test must be skipped

Returns `unittest2.skipIf`

Package Contents

Functions

<code>setting_is_set(option)</code>	Return either True or False if a Robottelo section setting is
<code>skip_if(cond, reason=None)</code>	Skips test if expected condition is True.
<code>skip_if_not_set(*options)</code>	Skips test if expected configuration is not set.
<code>stubbed(reason=None)</code>	Skips test due to non-implentation or some other reason.
<code>cacheable(func)</code>	Decorator that makes an optional object cache available
<code>run_only_on(project)</code>	Decorator to skip tests based on server mode.

`robottelo.decorators.settings`

`robottelo.decorators.NOT_IMPLEMENTED = Test not implemented`

`robottelo.decorators.LOGGER`

`robottelo.decorators.OBJECT_CACHE`

`robottelo.decorators.tier1`

`robottelo.decorators.tier2`

`robottelo.decorators.tier3`

`robottelo.decorators.tier4`

`robottelo.decorators.destructive`

`robottelo.decorators.upgrade`

`robottelo.decorators.run_in_one_thread`

`robottelo.decorators.parametrize`

`robottelo.decorators.fixture`

robottelo.decorators.**skipif**

robottelo.decorators.**setting_is_set** (*option*)

Return either True or False if a Robottelo section setting is set or not respectively.

robottelo.decorators.**skip_if** (*cond, reason=None*)

Skips test if expected condition is True.

Decorating a method:

```
@skip_if(foo is not bar, 'skipping due foo is not bar')
def test_something(self):
    self.assertTrue(True)
```

robottelo.decorators.**skip_if_not_set** (**options*)

Skips test if expected configuration is not set.

Decorating a method:

```
@skip_if_not_set('clients')
def test_something(self):
    self.assertTrue(True)
```

Decorating an entire class:

```
class FeatureTestCase(robottelo.test.TestCase):

    @skip_if_not_set('clients')
    def setUp(self):
        pass

    def test_something(self):
        self.assertTrue(True)
```

Or:

```
class FeatureTestCase(robottelo.test.TestCase):

    @classmethod
    @skip_if_not_set('clients')
    def setUpClass(cls):
        pass

    def test_something(self):
        self.assertTrue(True)
```

The last two approaches are required for decorating all test methods of a class, because the `skip_if_not_set` decorator is intended to run at runtime and not at import time. Decorating a class definition directly is not supported.

Be aware that nosetests and standard Python unittest runners are not able to identify the `SkipTest` exception being raised on `setUpClass` and will report a failure. On the other hand, pytest will handle this as expected.

Parameters *options* – List of valid *robottelo.properties* section names.

Raises `unittest2.SkipTest`: If expected configuration section is not fully set in the *robottelo.properties* file. All required attributes must be set. For example, if the *server* section is enabled but its *hostname* attribute is not set, then a test that expects it will be skipped.

robottelo.decorators.**stubbed** (*reason=None*)

Skips test due to non-implentation or some other reason.

`robottelo.decorators.cacheable` (*func*)
 Decorator that makes an optional object cache available

exception `robottelo.decorators.ProjectModeError`
 Bases: `Exception`

Indicates an error occurred while skipping based on Project Mode.

`robottelo.decorators.run_only_on` (*project*)
 Decorator to skip tests based on server mode.

If the calling function -

- uses 'sat' - test will be run for sat mode only
- uses 'sam' - test will be run for sam mode only
- does not use this decorator - test will be run for sat/sam modes

Note: The server mode is identified by `settings.project`.

Usage:

To skip a specific test:

```
from robottelo.decorators import run_only_on

@run_only_on('sat')
def test_hostgroup_create():
    # test code continues here
```

Parameters `project` (*str*) – Enter 'sat' for Satellite and 'sam' for SAM

Returns `unittest2.skipIf`

Raises `ProjectModeError()` if invalid *project* is given or invalid mode is specified in `robottelo.properties` file

`robottelo.report_portal`

Submodules

`robottelo.report_portal.portal`

Module Contents

Classes

<code>ReportPortal()</code>	Represents ReportPortal
<code>Launch(rp, launch_info)</code>	

`robottelo.report_portal.portal.LOGGER`

`robottelo.report_portal.portal.launch_types = ['satellite6', 'upgrades']`

class `robottelo.report_portal.portal.ReportPortal`
 Represents ReportPortal

This holds the properties and functions to interact with Report Portal properties and launches

defect_types

statuses = ['failed', 'passed', 'skipped', 'interrupted', 'in_progress']

api_url

Super url of report portal :returns: Base url for API request

headers

The headers for Report Portal Requests. :returns: header for API request

_format_launches (*self*, *launches*)

The pretty formatter function that formats launches in a structured way

Parameters **launches** (*filter*) – Satellite or Upgrade Type launches

Returns dict Launches, keyed with their snap_versions formatted as,
{‘snap_version1’:launch_object1, ‘snap_version2’: launch_object2}

_launch_requester (*self*)

The launch GET requester to fetch the all available launches of ReportPortal

Returns dict The json of all RP launches

launches (*self*, *sat_version=None*, *launch_type='satellite6'*)

Returns launches in Report Portal customized by *sat_version*, *launch_type* and latest number of launches sorted by latest sat version/snap version.

This does not includes each tests data for all tests in all launches, but it includes just an overview data for all and each launch in Report Portal

Parameters

- **sat_version** (*str*) – The satellite version If its not specified, then latest count of launches of satellite versions returned
- **launch_type** (*str*) – Either satellite6 or upgrades, default returns only non-upgrade launches

Returns dict The launches of Report portal. if *sat_version* is given, {‘snap_version1’:launch_object1, ‘snap_version2’:launch_object2} else, {‘sat_version1’:{‘snap_version1’:launch_..}, ‘sat_version2’:{}}

launch (*self*, *sat_version*, *snap_version=None*, *launch_type='satellite6'*)

Returns a specific launch data in Report Portal Project

This does not includes each tests data in launch

Parameters

- **sat_version** (*str*) – The satellite version
- **snap_version** (*str*) – The snap version of a given satellite version if None, the latest launch data of a given *sat_version* is returned
- **launch_type** (*str*) – Either satellite6 or upgrades, default returns only non-upgrade launches

Returns dict The data directory of requested or latest launch

class robottelo.report_portal.portal.**Launch** (*rp*, *launch_info*)

`_versions` (*self*)

Sets satellite and snap version attributes of a launch

`_test_params` (*self, status, defect_type*)

Customise parameters for Test items API request

Returns dict The parameters dict for API test items request

`_test_requester` (*self, params, page*)

The Test Items GET requester to fetch the data on a page

If any error and before Failing explicitly, it retries for 3 times with 10 seconds delay

Returns tuple (int, list) Total pages count and the list of tests along with each tests properties in a page

`tests` (*self, status=None, defect_type=None*)

Returns tests data customized by kwargs parameters.

This is a main function that will be called to retrieve the tests data of a particular test status or/and defect_type

Parameters

- **status** (*str*) – Filter tests of a launch with tests *status*
- **defect_type** (*str*) – Filter tests of a launch with tests *defect_type*

Returns dict All filtered tests dict based on params data keyed by test name and test properties as value, in format - `{'test_name1':test1_properties_dict, 'test_name2':test2_properties_dict}`

`robottelo.ui`

Submodules

`robottelo.ui.utils`

Module Contents

Functions

```
create_fake_host(session, host, interface_id=gen_string('alpha'),
                 global_parameters=None, host_parameters=None,
                 extra_values=None)
```

```
robottelo.ui.utils.create_fake_host(session, host, interface_id=gen_string('alpha'),
                                   global_parameters=None, host_parameters=None,
                                   extra_values=None)
```

`robottelo.utils`

Subpackages

`robottelo.utils.issue_handlers`

Submodules

`robottelo.utils.issue_handlers.bugzilla`

Module Contents

Functions

<code>is_open_bz(issue, data=None)</code>	Check if specific BZ is open consulting a cached <i>data</i> dict or
<code>should_deselect_bz(issue, data=None)</code>	Check if test should be deselected based on marked issue.
<code>follow_duplicates(bz)</code>	Recursively load the duplicate data
<code>extract_min_version(bz)</code>	return target_milestone or min(versions flags) or 0
<code>try_from_cache(issue, data=None)</code>	Try to fetch issue from given data cache or previous loaded on pytest.
<code>collect_data_bz(collected_data, cached_data)</code>	Collect data from BUgzilla API and aggregate in a dictionary.
<code>collect_dupes(bz, collected_data, cached_data=None)</code>	Recursively find for duplicates
<code>collect_clones(bz, collected_data, cached_data=None)</code>	Recursively find for clones.
<code>get_data_bz(bz_numbers, cached_data=None)</code>	Get a list of marked BZ data and query Bugzilla REST API.
<code>get_single_bz(number, cached_data=None)</code>	Call BZ API to get a single BZ data and cache it
<code>get_default_bz(number)</code>	This is the default BZ data when it is not possible to reach BZ api

`robottelo.utils.issue_handlers.bugzilla.LOGGER`

`robottelo.utils.issue_handlers.bugzilla.VERSION_RE`

`robottelo.utils.issue_handlers.bugzilla.is_open_bz (issue, data=None)`
 Check if specific BZ is open consulting a cached *data* dict or calling Bugzilla REST API.

Arguments: issue {str} – The BZ reference e.g: BZ:123456 data {dict} – Issue data indexed by <handler>:<number> or None

`robottelo.utils.issue_handlers.bugzilla.should_deselect_bz (issue, data=None)`
 Check if test should be deselected based on marked issue.

1. Resolution WONTFIX/CANTFIX/DEFERRED should deselect

Arguments: issue {str} – The BZ reference e.g: BZ:123456 data {dict} – Issue data indexed by <handler>:<number> or None

`robottelo.utils.issue_handlers.bugzilla.follow_duplicates (bz)`
 Recursively load the duplicate data

`robottelo.utils.issue_handlers.bugzilla.extract_min_version (bz)`
 return target_milestone or min(versions flags) or 0

`robottelo.utils.issue_handlers.bugzilla.try_from_cache (issue, data=None)`
 Try to fetch issue from given data cache or previous loaded on pytest.

Arguments: `issue {str}` – The BZ reference e.g: BZ:123456 `data {dict}` – Issue data indexed by `<handler>:<number>` or None

`robottelo.utils.issue_handlers.bugzilla.collect_data_bz` (*collected_data*,
cached_data)

Collect data from BUgzilla API and aggregate in a dictionary.

Arguments: `collected_data {dict}` – dict with BZs collected by pytest `cached_data {dict}` – Cached data previous loaded from API

`robottelo.utils.issue_handlers.bugzilla.collect_dupes` (*bz*, *collected_data*,
cached_data=None)

Recursively find for duplicates

`robottelo.utils.issue_handlers.bugzilla.collect_clones` (*bz*, *collected_data*,
cached_data=None)

Recursively find for clones. This handler does not process clones as part of skipping logic. but the data is fetched here to feed nagger script later.

`robottelo.utils.issue_handlers.bugzilla.CACHED_RESPONSES`

`robottelo.utils.issue_handlers.bugzilla.get_data_bz` (*bz_numbers*,
cached_data=None)

Get a list of marked BZ data and query Bugzilla REST API.

Arguments: `bz_numbers {list of str}` – [‘123456’, ...] `cached_data`

Returns: [list of dicts] – [{‘id’:..., ‘status’:..., ‘resolution’: ...}]

`robottelo.utils.issue_handlers.bugzilla.get_single_bz` (*number*, *cached_data=None*)

Call BZ API to get a single BZ data and cache it

`robottelo.utils.issue_handlers.bugzilla.get_default_bz` (*number*)

This is the default BZ data when it is not possible to reach BZ api

Package Contents

Functions

<code>add_workaround</code> (<i>data</i> , <i>matches</i> , <i>usage</i> , <i>validation=</i> <code>lambda *a, **k: True, **kwargs</code>)	Adds entry for workaround usage.
<code>should_deselect</code> (<i>issue</i> , <i>data=None</i>)	Check if test should be deselected based on marked issue.
<code>is_open</code> (<i>issue</i> , <i>data=None</i>)	Check if specific issue is open.

`robottelo.utils.issue_handlers.handler_methods`

`robottelo.utils.issue_handlers.SUPPORTED_HANDLERS`

`robottelo.utils.issue_handlers.add_workaround` (*data*, *matches*, *usage*, *validation=*`lambda *a, **k: True, **kwargs`)

Adds entry for workaround usage.

`robottelo.utils.issue_handlers.should_deselect` (*issue*, *data=None*)

Check if test should be deselected based on marked issue.

`robottelo.utils.issue_handlers.is_open` (*issue*, *data=None*)

Check if specific issue is open.

Issue must be prefixed by its handler e.g:

Bugzilla: BZ:123456

Arguments: `issue {str}` – A string containing handler + number e.g: BZ:123465 `data {dict}` – Issue data indexed by `<handler>:<number>` or `None`

Submodules

`robottelo.utils.version`

Module Contents

Classes

<code>VersionEncoder(*, skipkeys=False, ensure_ascii=True, check_circular=True, allow_nan=True, sort_keys=False, indent=None, separators=None, default=None)</code>	Transform Version instances to str
---	------------------------------------

Functions

<code>search_version_key(key, value)</code>	recursively look for 'version' key and transform it in a Version instance
---	---

`robottelo.utils.version.search_version_key(key, value)`
recursively look for 'version' key and transform it in a Version instance

class `robottelo.utils.version.VersionEncoder(*, skipkeys=False, ensure_ascii=True, check_circular=True, allow_nan=True, sort_keys=False, indent=None, separators=None, default=None)`

Bases: `json.JSONEncoder`

Transform Version instances to str

default (`self, z`)

Implement this method in a subclass such that it returns a serializable object for `o`, or calls the base implementation (to raise a `TypeError`).

For example, to support arbitrary iterators, you could implement default like this:

```
def default(self, o):
    try:
        iterable = iter(o)
    except TypeError:
        pass
    else:
        return list(iterable)
    # Let the base class default method raise the TypeError
    return JSONEncoder.default(self, o)
```

Submodules

robottelo.cleanup

Cleanup module for different entities

Module Contents**Functions**

<code>capsule_cleanup(proxy_id=None)</code>	Deletes the capsule with the given id
<code>realm_cleanup(realm_id=None)</code>	Deletes the realm with the given id
<code>location_cleanup(loc_id=None)</code>	Deletes the location with the given id
<code>org_cleanup(org_id=None)</code>	Deletes the Org with the given id
<code>host_cleanup(host_id=None)</code>	Deletes the Host with the given id
<code>setting_cleanup(setting_name=None, setting_value=None)</code>	Put necessary value for a specified setting
<code>vm_cleanup(vm)</code>	Destroys virtual machine
<code>cleanup_of_provisioned_server(hostname=None, provisioning_server=None, distro=None)</code>	Cleanup the VM from provisioning server

robottelo.cleanup.**LOGGER**

robottelo.cleanup.**capsule_cleanup** (*proxy_id=None*)
Deletes the capsule with the given id

robottelo.cleanup.**realm_cleanup** (*realm_id=None*)
Deletes the realm with the given id

robottelo.cleanup.**location_cleanup** (*loc_id=None*)
Deletes the location with the given id

robottelo.cleanup.**org_cleanup** (*org_id=None*)
Deletes the Org with the given id

robottelo.cleanup.**host_cleanup** (*host_id=None*)
Deletes the Host with the given id

robottelo.cleanup.**setting_cleanup** (*setting_name=None, setting_value=None*)
Put necessary value for a specified setting

robottelo.cleanup.**vm_cleanup** (*vm*)
Destroys virtual machine

Parameters *vm* (`robottelo.vm.VirtualMachine`) – virtual machine to destroy

robottelo.cleanup.**cleanup_of_provisioned_server** (*hostname=None, provisioning_server=None, distro=None*)

Cleanup the VM from provisioning server

Param str hostname: The content host hostname

Param str provisioning_server: provision server name

Param str distro: distro type

robottelo.constants

Defines various constants

Module Contents

```
robottelo.constants.LOCALES = ['ca', 'de', 'en', 'en_GB', 'es', 'fr', 'gl', 'it', 'ja', 'k  
robottelo.constants.DISTRO_RHEL6 = rhel6  
robottelo.constants.DISTRO_RHEL7 = rhel7  
robottelo.constants.DISTRO_RHEL8 = rhel8  
robottelo.constants.DISTRO_SLES11 = sles11  
robottelo.constants.DISTRO_SLES12 = sles12  
robottelo.constants.RHEL_6_MAJOR_VERSION = 6  
robottelo.constants.RHEL_7_MAJOR_VERSION = 7  
robottelo.constants.RHEL_8_MAJOR_VERSION = 8  
robottelo.constants.DISTRO_DEFAULT  
robottelo.constants.DISTROS_SUPPORTED  
robottelo.constants.DISTROS_MAJOR_VERSION  
robottelo.constants.MAJOR_VERSION_DISTRO  
robottelo.constants.INTERFACE_API = API  
robottelo.constants.INTERFACE_CLI = CLI  
robottelo.constants.FOREMAN_PROVIDERS  
robottelo.constants.EC2_REGION_CA_CENTRAL_1 = ca-central-1  
robottelo.constants.CONTENT_CREDENTIALS_TYPES  
robottelo.constants.VIRT_WHO_HYPERVISOR_TYPES  
robottelo.constants.LIBVIRT_RESOURCE_URL = qemu+ssh://root@%s/system  
robottelo.constants.RHEV_CR = %s (RHV)  
robottelo.constants.AWS_EC2_FLAVOR_T2_MICRO = t2.micro - T2 Micro Instance  
robottelo.constants.COMPUTE_PROFILE_LARGE = 3-Large  
robottelo.constants.COMPUTE_PROFILE_SMALL = 1-Small  
robottelo.constants._bcds  
robottelo.constants._abcfs  
robottelo.constants._abcs  
robottelo.constants._zones_combo  
robottelo.constants.VALID_GCE_ZONES  
robottelo.constants.LATEST_RHEL7_GCE_IMG_UUID = 7726764279310511390  
robottelo.constants.GCE_MACHINE_TYPE_DEFAULT = f1-micro
```



```
robottelo.constants.GCE_NETWORK_DEFAULT = default
robottelo.constants.GCE_EXTERNAL_IP_DEFAULT = True
robottelo.constants.AZURERM_VALID_REGIONS = ['East Asia', 'Southeast Asia', 'Central US',
robottelo.constants.AZURERM_RHEL7_FT_IMG_URN = marketplace://RedHat:RHEL:7-RAW:latest
robottelo.constants.AZURERM_RHEL7_UD_IMG_URN = marketplace://RedHat:RHEL:7-RAW-CI:7.6.2019
robottelo.constants.AZURERM_RHEL7_FT_BYOS_IMG_URN = marketplace://RedHat:rhel-byos:rhel-lvr
robottelo.constants.AZURERM_RHEL7_FT_CUSTOM_IMG_URN = custom://vm1-shared-image-2020051408
robottelo.constants.AZURERM_RHEL7_FT_GALLERY_IMG_URN = gallery://RHEL77img
robottelo.constants.AZURERM_RG_DEFAULT = SATQE
robottelo.constants.AZURERM_PLATFORM_DEFAULT = Linux
robottelo.constants.AZURERM_VM_SIZE_DEFAULT = Standard_B2ms
robottelo.constants.AZURERM_PREMIUM_OS_Disk = True
robottelo.constants.AZURERM_FILE_URI = https://raw.githubusercontent.com/SatelliteQE/robott
robottelo.constants.HTML_TAGS = ['A', 'ABBR', 'ACRONYM', 'ADDRESS', 'APPLET', 'AREA', 'B',
robottelo.constants.OPERATING_SYSTEMS
robottelo.constants.TEMPLATE_TYPES = ['finish', 'iPXE', 'provision', 'PXEGrub', 'PXELinux',
robottelo.constants.RESOURCE_DEFAULT = Bare Metal
robottelo.constants.OS_TEMPLATE_DATA_FILE = os_template.txt
robottelo.constants.DOMAIN = lab.dom.%s.com
robottelo.constants.PARTITION_SCRIPT_DATA_FILE = partition_script.txt
robottelo.constants.SNIPPET_DATA_FILE = snippet.txt
robottelo.constants.SNIPPET_URL = https://gist.github.com/sghai/8434467/raw
robottelo.constants.INSTALL_MEDIUM_URL = http://mirror.fakeos.org/%s/$major.$minor/os/$arch
robottelo.constants.VALID_GPG_KEY_FILE = valid_gpg_key.txt
robottelo.constants.ZOO_CUSTOM_GPG_KEY = zoo_custom_gpgkey.txt
robottelo.constants.VALID_GPG_KEY_BETA_FILE = valid_gpg_key_beta.txt
robottelo.constants.KEY_CLOAK_CLI = /opt/rh/rh-ss07/root/usr/share/keycloak/bin/kcadm.sh
robottelo.constants.RPM_TO_UPLOAD = which-2.19-6.el6.x86_64.rpm
robottelo.constants.SRPM_TO_UPLOAD = which-2.19-6.el6.src.rpm
robottelo.constants.ENVIRONMENT = Library
robottelo.constants.NOT_IMPLEMENTED = Test not implemented
robottelo.constants.SYNC_INTERVAL
robottelo.constants.REPO_TYPE
robottelo.constants.DOWNLOAD_POLICIES
robottelo.constants.CHECKSUM_TYPE
robottelo.constants.HASH_TYPE
```

```
robottelo.constants.REPO_TAB
robottelo.constants.PRDS
robottelo.constants.REPOSET
robottelo.constants.NO_REPOS_AVAILABLE = This system has no repositories available through
robottelo.constants.SM_OVERALL_STATUS
robottelo.constants.REPOS
robottelo.constants.DISTRO_REPOS
robottelo.constants.RHVA_REPO_TREE = [['rhel', 'rhva6', 'rhva65', 'repo_name', 'Red Hat Ent
robottelo.constants.SAT6_TOOLS_TREE = [['rhel', 'rhst6', 'rhst6', 'repo_name', 'Red Hat Sat
robottelo.constants.ATOMIC_HOST_TREE = [['rhah', 'rhaht', 'rhaht', 'repo_name', 'Red Hat E
robottelo.constants.DEFAULT_ORG = Default Organization
robottelo.constants.DEFAULT_LOC = Default Location
robottelo.constants.DEFAULT_CV = Default Organization View
robottelo.constants.DEFAULT_TEMPLATE = Kickstart default
robottelo.constants.DEFAULT_PXE_TEMPLATE = Kickstart default PXELinux
robottelo.constants.DEFAULT_ATOMIC_TEMPLATE = Atomic Kickstart default
robottelo.constants.DEFAULT_PTABLE = Kickstart default
robottelo.constants.DEFAULT_SUBSCRIPTION_NAME = Red Hat Enterprise Linux Server, Premium (I
robottelo.constants.DEFAULT_ARCHITECTURE = x86_64
robottelo.constants.DEFAULT_RELEASE_VERSION = 6Server
robottelo.constants.DEFAULT_ROLE = Default role
robottelo.constants.LANGUAGES
robottelo.constants.SATELLITE_SUBSCRIPTION_NAME = Red Hat Satellite Infrastructure Subscrip
robottelo.constants.SATELLITE_FIREWALL_SERVICE_NAME = RH-Satellite-6
robottelo.constants.VDC_SUBSCRIPTION_NAME = Red Hat Enterprise Linux for Virtual Datacenter
robottelo.constants.TIMEZONES = ['(GMT+00:00) UTC', '(GMT-10:00) Hawaii', '(GMT+02:00) Kyiv
robottelo.constants.FILTER_CONTENT_TYPE
robottelo.constants.FILTER_TYPE
robottelo.constants.FILTER_ERRATA_TYPE
robottelo.constants.FILTER_ERRATA_DATE
robottelo.constants.REPORT_TEMPLATE_FILE = report_template.txt
robottelo.constants.REP_TEM_APPLIED_ERRATA_INPUT
robottelo.constants.DOCKER_REGISTRY_HUB = https://registry-1.docker.io
robottelo.constants.DOCKER_UPSTREAM_NAME = busybox
robottelo.constants.DOCKER_RH_REGISTRY_UPSTREAM_NAME = openshift3/ose-metrics-hawkular-oper
robottelo.constants.CUSTOM_FILE_REPO = https://fixtures.pulpproject.org/file/
```

```
robottelo.constants.CUSTOM_LOCAL_FOLDER = /var/www/html/myrepo/
robottelo.constants.CUSTOM_LOCAL_FILE = /var/www/html/myrepo/test.txt
robottelo.constants.CUSTOM_FILE_REPO_FILES_COUNT = 3
robottelo.constants.CUSTOM_KICKSTART_REPO = http://mirror.linux.duke.edu/pub/centos/8/Base
robottelo.constants.CUSTOM_RPM_REPO = https://fixtures.pulpproject.org/rpm-signed/
robottelo.constants.CUSTOM_RPM_SHA_512 = https://fixtures.pulpproject.org/rpm-with-sha-512
robottelo.constants.CUSTOM_RPM_SHA_512_FEED_COUNT
robottelo.constants.CUSTOM_MODULE_STREAM_REPO_1 = https://partha.fedorapeople.org/test-repo
robottelo.constants.CUSTOM_MODULE_STREAM_REPO_2 = https://partha.fedorapeople.org/test-repo
robottelo.constants.CUSTOM_SWID_TAG_REPO = https://partha.fedorapeople.org/test-repos/swid-
robottelo.constants.CUSTOM_REPODATA_PATH = /var/lib/pulp/published/yum/https/repos
robottelo.constants.CERT_PATH = /etc/pki/ca-trust/source/anchors/
robottelo.constants.FAKE_0_YUM_REPO = http://inecas.fedorapeople.org/fakerepos/zoo/
robottelo.constants.FAKE_1_YUM_REPO = http://inecas.fedorapeople.org/fakerepos/zoo3/
robottelo.constants.FAKE_2_YUM_REPO = http://inecas.fedorapeople.org/fakerepos/zoo2/
robottelo.constants.FAKE_3_YUM_REPO = http://omaciel.fedorapeople.org/fakerepo01
robottelo.constants.FAKE_4_YUM_REPO = http://omaciel.fedorapeople.org/fakerepo02
robottelo.constants.FAKE_5_YUM_REPO = http://{0}:{1}@rplevka.fedorapeople.org/fakerepo01/
robottelo.constants.FAKE_6_YUM_REPO = https://stephenw.fedorapeople.org/fakerepos/needed-e
robottelo.constants.FAKE_7_YUM_REPO = https://fixtures.pulpproject.org/rpm-long-updateinfo
robottelo.constants.FAKE_8_YUM_REPO = https://abalakht.fedorapeople.org/test_repos/lots_fi
robottelo.constants.FAKE_9_YUM_REPO = https://stephenw.fedorapeople.org/fakerepos/multiple
robottelo.constants.FAKE_10_YUM_REPO = https://partha.fedorapeople.org/test-repos/separate
robottelo.constants.FAKE_11_YUM_REPO = https://partha.fedorapeople.org/test-repos/separate
robottelo.constants.FAKE_YUM_DRPM_REPO = https://fixtures.pulpproject.org/drpm-signed/
robottelo.constants.FAKE_YUM_SRPM_REPO = https://fixtures.pulpproject.org/srpm-signed/
robottelo.constants.FAKE_YUM_SRPM_DUPLICATE_REPO = https://fixtures.pulpproject.org/srpm-d
robottelo.constants.FAKE_YUM_MIXED_REPO = https://pondrejk.fedorapeople.org/test_repos/mix
robottelo.constants.FAKE_0_YUM_REPO_PACKAGES_COUNT = 32
robottelo.constants.CUSTOM_PUPPET_REPO = http://omaciel.fedorapeople.org/bagoftricks
robottelo.constants.FAKE_0_PUPPET_REPO = http://davidd.fedorapeople.org/repos/random_puppet
robottelo.constants.FAKE_1_PUPPET_REPO = http://omaciel.fedorapeople.org/fakepuppet01
robottelo.constants.FAKE_2_PUPPET_REPO = http://omaciel.fedorapeople.org/fakepuppet02
robottelo.constants.FAKE_3_PUPPET_REPO = http://omaciel.fedorapeople.org/fakepuppet03
robottelo.constants.FAKE_4_PUPPET_REPO = http://omaciel.fedorapeople.org/fakepuppet04
robottelo.constants.FAKE_5_PUPPET_REPO = http://omaciel.fedorapeople.org/fakepuppet05
```

```
robottelo.constants.FAKE_6_PUPPET_REPO = http://kbidarka.fedorapeople.org/repos/puppet-modu
robottelo.constants.FAKE_7_PUPPET_REPO = http://{0}:{1}@rplevka.fedorapeople.org/fakepuppet
robottelo.constants.FAKE_8_PUPPET_REPO = https://omaciel.fedorapeople.org/f4cb00ed/
robottelo.constants.FEDORA26_OSTREE_REPO = https://kojipkgs.fedoraproject.org/compose/ostree
robottelo.constants.FEDORA27_OSTREE_REPO = https://kojipkgs.fedoraproject.org/compose/ostree
robottelo.constants.REPO_DISCOVERY_URL = http://omaciel.fedorapeople.org/
robottelo.constants.FAKE_0_INC_UPD_URL = https://abalakht.fedorapeople.org/test_files/inc_
robottelo.constants.FAKE_0_INC_UPD_ERRATA = EXA:2015-0002
robottelo.constants.FAKE_0_INC_UPD_OLD_PACKAGE = pulp-test-package-0.2.1-1.fc11.x86_64.rpm
robottelo.constants.FAKE_0_INC_UPD_NEW_PACKAGE = pulp-test-package-0.3.1-1.fc11.x86_64.rpm
robottelo.constants.FAKE_0_INC_UPD_OLD_UPDATEFILE = updateinfo.xml
robottelo.constants.FAKE_0_INC_UPD_NEW_UPDATEFILE = updateinfo_v2.xml
robottelo.constants.INVALID_URL = http://username:password@@example.com/repo
robottelo.constants.FAKE_0_CUSTOM_PACKAGE = bear-4.1-1.noarch
robottelo.constants.FAKE_0_CUSTOM_PACKAGE_NAME = bear
robottelo.constants.FAKE_1_CUSTOM_PACKAGE = walrus-0.71-1.noarch
robottelo.constants.FAKE_1_CUSTOM_PACKAGE_NAME = walrus
robottelo.constants.FAKE_2_CUSTOM_PACKAGE = walrus-5.21-1.noarch
robottelo.constants.FAKE_2_CUSTOM_PACKAGE_NAME = walrus
robottelo.constants.FAKE_3_CUSTOM_PACKAGE = duck-0.8-1.noarch
robottelo.constants.FAKE_3_CUSTOM_PACKAGE_NAME = duck
robottelo.constants.FAKE_4_CUSTOM_PACKAGE = kangaroo-0.2-1.noarch
robottelo.constants.FAKE_4_CUSTOM_PACKAGE_NAME = kangaroo
robottelo.constants.FAKE_5_CUSTOM_PACKAGE = kangaroo-0.3-1.noarch
robottelo.constants.REAL_0_RH_PACKAGE = rhvm-sdk-python-3.3.0.21-1.el6ev.noarch
robottelo.constants.REAL_RHEL7_0_0_PACKAGE = liblouis-python-2.5.2-10.el7.noarch
robottelo.constants.REAL_RHEL7_0_0_PACKAGE_NAME = liblouis-python
robottelo.constants.REAL_RHEL7_0_1_PACKAGE = liblouis-python-2.5.2-11.el7_4.noarch
robottelo.constants.REAL_RHEL7_0_1_PACKAGE_FILENAME = liblouis-python-2.5.2-11.el7_4.noarch
robottelo.constants.FAKE_0_CUSTOM_PACKAGE_GROUP_NAME = birds
robottelo.constants.FAKE_9_YUM_OUTDATED_PACKAGES = ['bear-4.0-1.noarch', 'crow-0.7-1.noarch']
robottelo.constants.FAKE_9_YUM_UPDATED_PACKAGES = ['bear-4.1-1.noarch', 'crow-0.8-1.noarch']
robottelo.constants.FAKE_0_MODULAR_ERRATA_ID = RHEA-2012:0059
robottelo.constants.FAKE_0_ERRATA_ID = RHEA-2012:0001
robottelo.constants.FAKE_1_ERRATA_ID = RHEA-2012:0002
robottelo.constants.FAKE_2_ERRATA_ID = RHSA-2012:0055
```

```
robottelo.constants.FAKE_3_ERRATA_ID = RHEA-2012:7269
robottelo.constants.FAKE_4_ERRATA_ID = WALRUS-2013:0002
robottelo.constants.FAKE_5_ERRATA_ID = RHBA-2012:1030
robottelo.constants.REAL_0_ERRATA_ID = RHBA-2019:3175
robottelo.constants.REAL_1_ERRATA_ID = RHBA-2016:1357
robottelo.constants.REAL_2_ERRATA_ID = RHEA-2014:0657
robottelo.constants.REAL_4_ERRATA_ID = RHSA-2014:1873
robottelo.constants.REAL_4_ERRATA_CVES = ['CVE-2014-3633', 'CVE-2014-3657', 'CVE-2014-7823']
robottelo.constants.REAL_RHEL7_0_ERRATA_ID = RHSA-2017:3111
robottelo.constants.REAL_RHEL7_1_ERRATA_ID = RHBA-2017:0395
robottelo.constants.FAKE_0_YUM_ERRATUM_COUNT = 4
robottelo.constants.FAKE_1_YUM_ERRATUM_COUNT = 4
robottelo.constants.FAKE_1_YUM_REPOS_COUNT = 32
robottelo.constants.FAKE_3_YUM_ERRATUM_COUNT = 79
robottelo.constants.FAKE_3_YUM_REPOS_COUNT = 136
robottelo.constants.FAKE_6_YUM_ERRATUM_COUNT = 5
robottelo.constants.FAKE_9_YUM_ERRATUM_COUNT = 5
robottelo.constants.FAKE_9_YUM_ERRATUM = ['RHSA-2012:0055', 'RHSA-2012:0056', 'RHSA-2012:0057']
robottelo.constants.FAKE_9_YUM_SECURITY_ERRATUM = ['RHSA-2012:0055', 'RHSA-2012:0056', 'RHSA-2012:0057']
robottelo.constants.FAKE_9_YUM_SECURITY_ERRATUM_COUNT = 3
robottelo.constants.FAKE_10_YUM_BUGFIX_ERRATUM = ['RHBA-2012:1030']
robottelo.constants.FAKE_10_YUM_BUGFIX_ERRATUM_COUNT = 1
robottelo.constants.FAKE_11_YUM_ENHANCEMENT_ERRATUM = ['RHEA-2012:0058']
robottelo.constants.FAKE_11_YUM_ENHANCEMENT_ERRATUM_COUNT = 1
robottelo.constants.PUPPET_MODULE_NTP_PUPPETLABS = puppetlabs-ntp-3.2.1.tar.gz
robottelo.constants.PUPPET_MODULE_CUSTOM_FILE_NAME = puppet_custom_selinux-0.3.1.tar.gz
robottelo.constants.PUPPET_MODULE_CUSTOM_NAME = selinux
robottelo.constants.FAKE_0_CUSTOM_PACKAGE_GROUP = ['cockateel-3.1-1.noarch', 'duck-0.6-1.noarch']
robottelo.constants.FAKE_1_YUM_REPO_RPMS = ['bear-4.1-1.noarch.rpm', 'camel-0.1-1.noarch.rpm']
robottelo.constants.FAKE_0_PUPPET_MODULE = httpd
robottelo.constants.FAKE_PULP_REMOTE_FILEREPO = https://pondrejk.fedorapeople.org/test_repos/
robottelo.constants.FAKE_0_YUM_REPO_STRING_BASED_VERSIONS = https://fixtures.pulpproject.org/
robottelo.constants.FAKE_0_YUM_REPO_STRING_BASED_VERSIONS_COUNTS = {}
robottelo.constants.PULP_PUBLISHED_ISO_REPOS_PATH = /var/lib/pulp/published/http/isos
robottelo.constants.PULP_PUBLISHED_PUPPET_REPOS_PATH = /var/lib/pulp/published/puppet/http/repos
robottelo.constants.PULP_PUBLISHED_YUM_REPOS_PATH = /var/lib/pulp/published/yum/http/repos
```

```
robottelo.constants.PERMISSIONS
robottelo.constants.PERMISSIONS_UI
robottelo.constants.ANY_CONTEXT
robottelo.constants.SUBNET_IPAM_TYPES
robottelo.constants.TREND_TYPES
robottelo.constants.LDAP_SERVER_TYPE
robottelo.constants.LDAP_ATTR
robottelo.constants.OSCAP_PERIOD
robottelo.constants.OSCAP_WEEKDAY
robottelo.constants.OSCAP_DEFAULT_CONTENT
robottelo.constants.OSCAP_PROFILE
robottelo.constants.ROLES = ['Access Insights Admin', 'Access Insights Viewer', 'Ansible R
robottelo.constants.ROLES_UNLOCKED = ['Access Insights Admin', 'Access Insights Viewer', '
robottelo.constants.ROLES_LOCKED = ['Discovery Manager', 'Discovery Reader', 'Edit hosts',
robottelo.constants.BOOKMARK_ENTITIES
robottelo.constants.STRING_TYPES = ['alpha', 'numeric', 'alphanumeric', 'latin1', 'utf8',
robottelo.constants.REAL_4_ERRATA_DETAILS = [None, None, ['Type', 'Security Advisory'], ['S
robottelo.constants.TOOLS_ERRATA_DETAILS = [['Advisory', 'RHBA-2016:1503'], ['CVEs', 'N/A']
robottelo.constants.TOOLS_ERRATA_TABLE_DETAILS = ['RHBA-2016:1503', 'Satellite 6.2 Tools Re
robottelo.constants.BACKUP_FILES = ['config_files.tar.gz', '.config.snar', 'metadata.yml',
robottelo.constants.HOT_BACKUP_FILES = ['candlepin.dump', 'config_files.tar.gz', '.config.s
robottelo.constants.VMWARE_CONSTANTS
robottelo.constants.HAMMER_CONFIG = ~/.hammer/cli.modules.d/foreman.yml
robottelo.constants.ANSWERS = /etc/foreman-installer/scenarios.d/satellite-answers.yaml
robottelo.constants.FOREMAN_TEMPLATE_IMPORT_URL = https://github.com/SatelliteQE/foreman_t
robottelo.constants.FOREMAN_TEMPLATES_COMMUNITY_URL = https://github.com/theforeman/commu
robottelo.constants.FOREMAN_TEMPLATE_TEST_TEMPLATE = https://raw.githubusercontent.com/Sate
robottelo.constants.FOREMAN_TEMPLATE_ROOT_DIR = /usr/share/foreman_templates
robottelo.constants.DEFAULT_SYSPURPOSE_ATTRIBUTES
robottelo.constants.OPEN_STATUSES = ['NEW', 'ASSIGNED', 'POST', 'MODIFIED']
robottelo.constants.CLOSED_STATUSES = ['ON_QA', 'VERIFIED', 'RELEASE_PENDING', 'CLOSED']
robottelo.constants.WONTFIX_RESOLUTIONS = ['WONTFIX', 'CANTFIX', 'DEFERRED']
robottelo.constants.GROUP_MEMBERSHIP_MAPPER
robottelo.constants.AUDIENCE_MAPPER
robottelo.constants.RHSSO_NEW_USER
robottelo.constants.RHSSO_RESET_PASSWORD
```

```
robottelo.constants.FOREMAN_ANSIBLE_MODULES = ['foreman_architecture', 'foreman_auth_source
```

```
robottelo.constants.FAM_MODULE_PATH = /usr/share/ansible/collections/ansible_collections/r
```

robottelo.datafactory

Data Factory for all entities

Module Contents

Functions

<code>filtered_datapoint(func)</code>	Overrides the data creator functions in this class to return 1 value and
<code>parametrized(data)</code>	Transforms data dictionary to pytest's parametrize acceptable format.
<code>generate_strings_list(length=None, exclude_types=None, min_length=3, max_length=30)</code>	Generates a list of different input strings.
<code>add_uppercase_char_into_string(text=None, length=10)</code>	Fix string to include a minimum of one uppercase character.
<code>invalid_emails_list()</code>	Returns a list of invalid emails.
<code>invalid_boolean_strings(list_len=10)</code>	Create a list of invalid booleans. E.g not true nor false
<code>xdist_adapter(argvalues)</code>	Adapter to avoid error when running tests on xdist
<code>invalid_id_list()</code>	Generates a list of invalid IDs.
<code>invalid_names_list()</code>	Generates a list of invalid names.
<code>valid_domain_names(interface=None, length=None)</code>	Valid domain names.
<code>invalid_domain_names(interface=None)</code>	Invalid domain names.
<code>invalid_usernames_list()</code>	
<code>invalid_values_list(interface=None)</code>	Generates a list of invalid input values.
<code>valid_data_list(interface=None)</code>	Generates a list of valid input values.
<code>valid_docker_repository_names()</code>	Generates a list of valid names for Docker repository.
<code>valid_emails_list()</code>	Returns a list of valid emails.
<code>valid_environments_list()</code>	Returns a list of valid environment names
<code>valid_hosts_list(domain_length=10)</code>	Generates a list of valid host names.
<code>valid_hostgroups_list()</code>	Generates a list of valid host group names.
<code>valid_labels_list()</code>	Generates a list of valid labels.
<code>valid_names_list()</code>	Generates a list of valid names.
<code>valid_org_names_list()</code>	Generates a list of valid organization names.
<code>valid_usernames_list()</code>	Returns a list of valid user names.
<code>valid_interfaces_list()</code>	Generates a list of valid host interface names.
<code>invalid_interfaces_list()</code>	Generates a list of invalid host interface names.
<code>valid_http_credentials(url_encoded=False)</code>	Returns a list of valid credentials for HTTP authentication
<code>invalid_http_credentials(url_encoded=False)</code>	Returns a list of invalid credentials for HTTP authentication
<code>invalid_docker_upstream_names()</code>	Return a list of various kinds of invalid strings for Docker
<code>valid_docker_upstream_names()</code>	Return a list of various kinds of valid strings for Docker repositories.
<code>valid_url_list()</code>	
<code>valid_cron_expressions()</code>	Returns a list of valid cron expressions

exception `robottelo.datafactory.InvalidArgumentError`

Bases: `Exception`

Indicates an error when an invalid argument is received.

`robottelo.datafactory.filtered_datapoint` (*func*)

Overrides the data creator functions in this class to return 1 value and transforms data dictionary to pytest's parametrize acceptable format for new style generators.

If `run_one_datapoint=false`, return the entire data set. (default: `False`) If `run_one_datapoint=true`, return a random data.

`robottelo.datafactory.parametrized` (*data*)

Transforms data dictionary to pytest's parametrize acceptable format. Generates parametrized test names from data dict keys.

Parameters `data` (*dict*) – dictionary with parametrized test names as dict keys and parametrized arguments as dict values

`robottelo.datafactory.generate_strings_list` (*length=None, exclude_types=None, min_length=3, max_length=30*)

Generates a list of different input strings.

Parameters

- **length** (*int*) – Specifies the length of the strings to be generated. If the len1 is None then the list is returned with string types of random length.
- **exclude_types** – Specify a list of data types to be removed from generated list. example: `exclude_types=['html', 'cjk']`
- **min_length** (*int*) – Minimum length to be used in integer generator
- **max_length** (*int*) – Maximum length to be used in integer generator

Returns A list of various string types.

`robottelo.datafactory.add_uppercase_char_into_string` (*text=None, length=10*)

Fix string to include a minimum of one uppercase character. <https://github.com/SatelliteQE/robottelo/issues/4742>

Parameters

- **text** (*str*) – String to include uppercase character.
- **length** (*int*) – Length of string that we create in case string to change was not provided.

`robottelo.datafactory.invalid_emails_list` ()

Returns a list of invalid emails.

Based on RFC 5321 and 5322, however consecutive dots are removed from the list, as such emails, e.g. `email@example.c` or `dot.dot@example.com` are common on the wild and it was decided to treat them as valid.

For more information, see [Bugzilla #1455501](#).

`robottelo.datafactory.invalid_boolean_strings` (*list_len=10*)

Create a list of invalid booleans. E.g not true nor false

Parameters `list_len` – len of the list to be generated

Returns list

`robottelo.datafactory.xdist_adapter` (*argvalues*)

Adapter to avoid error when running tests on xdist Check <https://github.com/pytest-dev/pytest-xdist/issues/149>

It returns a dict with `lst` as `argvalues` and `range(len(lst))` as `ids`

Since every run has the same number of values, ids is going to be the same on different workers.

```
dct = xdist_adapter(invalid_boolean_strings())

@pytest.mark.parametrize('value', **dct)
def test_something(value):
    #some code here
```

Parameters `argvalues` – to be passed to `parametrize`

Returns dict

`robottelo.datafactory.invalid_id_list()`

Generates a list of invalid IDs.

`robottelo.datafactory.invalid_names_list()`

Generates a list of invalid names.

`robottelo.datafactory.valid_domain_names(interface=None, length=None)`

Valid domain names.

`robottelo.datafactory.invalid_domain_names(interface=None)`

Invalid domain names.

`robottelo.datafactory.invalid_usernames_list()`

`robottelo.datafactory.invalid_values_list(interface=None)`

Generates a list of invalid input values.

This returns invalid values from `invalid_names_list()` and some interface (api/cli/ui) specific empty string values.

Parameters `interface` (*str*) – Interface name (one of api/cli/ui).

Returns Returns the invalid values list

Raises `InvalidArgumentError()`: If an invalid interface is received.

`robottelo.datafactory.valid_data_list(interface=None)`

Generates a list of valid input values.

Note: Although this helper is widely used for different attributes for several entities, the following are known behaviors and are handled specifically in the corresponding test modules:

```
Org - name max length is 242
Loc - name max length is 246
```

`robottelo.datafactory.valid_docker_repository_names()`

Generates a list of valid names for Docker repository.

`robottelo.datafactory.valid_emails_list()`

Returns a list of valid emails.

`robottelo.datafactory.valid_environments_list()`

Returns a list of valid environment names

`robottelo.datafactory.valid_hosts_list(domain_length=10)`

Generates a list of valid host names.

Note:: Host name format stored in db is 'fqdn=' + host_name + '.' + domain_name Host name max length is: 255 - 'fqdn=' - '.' - domain name length (default is 10) = 239 chars (by default). Name should be transformed into lower case

Parameters `domain_length` (*int*) – Domain name length (default is 10).

Returns Returns the valid host names list

`robottelo.datafactory.valid_hostgroups_list ()`

Generates a list of valid host group names.

Note:: Host group name max length is 245 chars. 220 chars for html as the largest html tag in fauxfactory is 10 chars long, so 245 - (10 chars + 10 chars + '<></>' chars) = 220 chars.

Returns Returns the valid host group names list

`robottelo.datafactory.valid_labels_list ()`

Generates a list of valid labels.

`robottelo.datafactory.valid_names_list ()`

Generates a list of valid names.

`robottelo.datafactory.valid_org_names_list ()`

Generates a list of valid organization names.

Note:: Organization name max length is 242 chars. 217 chars for html as the largest html tag in fauxfactory is 10 chars long, so 242 - (10 chars + 10 chars + '<></>' chars) = 217 chars.

Returns Returns the valid organization names list

`robottelo.datafactory.valid_usernames_list ()`

Returns a list of valid user names.

`robottelo.datafactory.valid_interfaces_list ()`

Generates a list of valid host interface names.

`robottelo.datafactory.invalid_interfaces_list ()`

Generates a list of invalid host interface names.

`robottelo.datafactory.valid_http_credentials (url_encoded=False)`

Returns a list of valid credentials for HTTP authentication The credentials dictionary contains the following keys: login - a username pass - a password quote - a Bool flag stating whether the credentials include special chars http_valid - a Bool flag stating whether the HTTP authentication will pass successfully on the server

Parameters `url_encoded` – flag for quoting special characters

Returns A list of dictionaries with user and password credentials

`robottelo.datafactory.invalid_http_credentials (url_encoded=False)`

Returns a list of invalid credentials for HTTP authentication

Parameters `url_encoded` – flag for quoting special characters

Returns A list of dictionaries with user and password credentials

`robottelo.datafactory.invalid_docker_upstream_names ()`

Return a list of various kinds of invalid strings for Docker repositories.

`robottelo.datafactory.valid_docker_upstream_names ()`

Return a list of various kinds of valid strings for Docker repositories.

`robottelo.datafactory.valid_url_list ()`

`robottelo.datafactory.valid_cron_expressions ()`

Returns a list of valid cron expressions

robottelo.errors

Custom Errors for Robottelo

Module Contents

exception `robottelo.errors.GCECertNotFound`

Bases: `Exception`

An exception to raise when GCE Cert json is not available for creating GCE CR

exception `robottelo.errors.TemplateNotFound`

Bases: `Exception`

An exception to raise when Template is not available in Satellite

robottelo.helpers

Several helper methods and functions.

Module Contents

Classes

<code>ServerFileDownloader()</code>	Downloads file from given fileurl to local /temp directory.
<code>Storage(*args, **kwargs)</code>	Turns a dict into an attribute based object.

Functions

<code>file_downloader(file_url, file_name=None, hostname=None, local_path=None)</code>	Downloads file from given fileurl to directory specified by local_path
<code>get_server_software()</code>	Figure out which product distribution is installed on the server.
<code>get_server_version()</code>	Read Satellite version.
<code>get_host_info(hostname=None)</code>	Get remote host's distribution information
<code>get_nailgun_config(user=None)</code>	Return a NailGun configuration file constructed from default values.
<code>escape_search(term)</code>	Wraps a search term in " and escape term's " and characters
<code>update_dictionary(default, updates)</code>	Updates default dictionary with elements from
<code>get_data_file(filename)</code>	Returns correct path of file from data folder.
<code>read_data_file(filename)</code>	Read the contents of data file
<code>install_katello_ca(hostname=None)</code>	Downloads and installs katello-ca rpm
<code>remove_katello_ca(hostname=None)</code>	Removes katello-ca rpm
<code>md5_by_url(url, hostname=None)</code>	Returns md5 checksum of a file, accessible via URL. Useful when you want
<code>add_remote_execution_ssh_key(hostname, key_path=None, proxy_hostname=None, **kwargs)</code>	Add remote execution keys to the client
<code>get_available_capsule_port(port_pool=None)</code>	returns a list of unused ports dedicated for fake capsules

Continued on next page

Table 95 – continued from previous page

<code>default_url_on_new_port(oldport, newport)</code>	Creates context where the default capsule is forwarded on a new port
<code>get_func_name(func, test_item=None)</code>	Given a func object return standardized name to use across project
<code>get_services_status()</code>	Check if core services are running
<code>form_repo_path(org=None, lce=None, cv=None, cvv=None, prod=None, repo=None, capsule=False)</code>	Forms unix path to the directory containing published repository in
<code>create_repo(name, repo_fetch_url=None, packages=None, wipe_repodata=False, hostname=None)</code>	Creates a repository from given packages and publishes it into pulp's
<code>repo_add_updateinfo(name, updateinfo_url=None, hostname=None)</code>	Modify repo with contents of updateinfo.xml file.
<code>extract_capsule_satellite_installer_command(text)</code>	Extract satellite installer command from capsule-certs-generate command
<code>extract_ui_token(input)</code>	Extracts and returns the CSRF protection token from a given
<code>get_web_session()</code>	Logs in as admin user and returns the valid requests.Session object
<code>host_provisioning_check(ip_addr)</code>	Check the provisioned host status by pinging the ip of host and check
<code>slugify_component(string, keep_hyphens=True)</code>	Make component name a slug
<code>download_gce_cert()</code>	
<code>idgen(val)</code>	The id generator function which will return string that will append to the parameterized

`robottelo.helpers`.**LOGGER**

exception `robottelo.helpers.DataFileError`

Bases: Exception

Indicates any issue when reading a data file.

exception `robottelo.helpers.HostInfoError`

Bases: Exception

Indicates any issue when getting host info.

exception `robottelo.helpers.ProvisioningCheckError`

Bases: Exception

Indicates any issue when provisioning a host.

exception `robottelo.helpers.InvalidArgumentError`

Bases: Exception

Indicates an error when an invalid argument is received.

exception `robottelo.helpers.ProxyError`

Bases: Exception

Indicates an error in state of proxy

exception `robottelo.helpers.DownloadFileError`

Bases: Exception

Indicates an error when failure in downloading file from server.

class `robottelo.helpers.ServerFileDownloader`

Bases: object

Downloads file from given fileurl to local /temp directory.

`__call__(self, extention, fileurl)`

Downloads file from given fileurl to local /temp directory with given extention.

Parameters

- **extention** (*str*) – The file extention with which the file to be saved in /temp directory.
- **fileurl** (*str*) – The complete server file path from where the file will be downloaded.

Returns Returns complete file path with name of downloaded file.

`robottelo.helpers.download_server_file`

`robottelo.helpers.file_downloader(file_url, local_path=None, file_name=None, hostname=None)`

Downloads file from given fileurl to directory specified by local_path with given file_name on host specified by hostname. Leave hostname as None to download file on the localhost. If remote directory is not specified it downloads file to /tmp/.

Parameters

- **file_url** (*str*) – The complete server file path from where the file will be downloaded.
- **local_path** (*str*) – Name of directory where file will be saved. If not provided file will be saved in /tmp/ directory.
- **file_name** (*str*) – Name of the file to be saved with. If not provided filename from url will be used.
- **hostname** (*str*) – Hostname of server where the file need to be downloaded.

Returns Returns list containing complete file path and name of downloaded file.

`robottelo.helpers.get_server_software()`

Figure out which product distribution is installed on the server.

Returns Either 'upstream' or 'downstream'.

Return type str

`robottelo.helpers.get_server_version()`

Read Satellite version.

Inspect server /usr/share/foreman/lib/satellite/version.rb in order to get the installed Satellite version.

Returns Either a string containing the Satellite version or None if the version.rb file is not present.

`robottelo.helpers.get_host_info(hostname=None)`

Get remote host's distribution information

Parameters **hostname** (*str*) – Hostname or IP address of the remote host. If None the hostname will be get from `main.server.hostname` config.

Returns A tuple in the form (distro, major, minor). major and minor are integers. minor can be None if not available.

`robottelo.helpers.get_nailgun_config(user=None)`

Return a NailGun configuration file constructed from default values.

Parameters **user** – The ``nailgun.entities.User`` object of an user with additional passwd property/attribute

Returns `nailgun.config.ServerConfig` object, populated from user parameter object else with values from `robottelo.config.settings`

`robottelo.helpers.escape_search` (*term*)

Wraps a search term in " and escape term's " and characters

`robottelo.helpers.update_dictionary` (*default, updates*)

Updates default dictionary with elements from optional dictionary.

@param default: A python dictionary containing the minimal required arguments to create a CLI object.

@param updates: A python dictionary containing attributes to overwrite on default dictionary.

@return default: The modified default python dictionary.

`robottelo.helpers.get_data_file` (*filename*)

Returns correct path of file from data folder.

`robottelo.helpers.read_data_file` (*filename*)

Read the contents of data file

`robottelo.helpers.install_katello_ca` (*hostname=None*)

Downloads and installs katello-ca rpm

Parameters `hostname` (*str*) – Hostname or IP address of the remote host. If `None` the hostname will be get from `main.server.hostname` config

Returns `None`.

Raises `AssertionError`: If katello-ca wasn't installed.

`robottelo.helpers.remove_katello_ca` (*hostname=None*)

Removes katello-ca rpm

Parameters `hostname` (*str*) – Hostname or IP address of the remote host. If `None` the hostname will be get from `main.server.hostname` config

Returns `None`.

Raises `AssertionError`: If katello-ca wasn't removed.

`robottelo.helpers.md5_by_url` (*url, hostname=None*)

Returns md5 checksum of a file, accessible via URL. Useful when you want to calculate checksum but don't want to deal with storing a file and removing it afterwards.

Parameters

- `url` (*str*) – URL of a file.
- `hostname` (*str*) – Hostname or IP address of the remote host. If `None` the hostname will be get from `main.server.hostname` config

Return str string containing md5 checksum.

Raises `AssertionError`: If non-zero return code received (file couldn't be reached or calculation was not successful).

`robottelo.helpers.add_remote_execution_ssh_key` (*hostname, key_path=None, proxy_hostname=None, **kwargs*)

Add remote execution keys to the client

Parameters

- `proxy_hostname` (*str*) – external capsule hostname
- `hostname` (*str*) – The client hostname
- `key` (*str*) – Path to a key on the satellite server
- `kwargs` (*dict*) – directly passed to `ssh.add_authorized_key`

`robottelo.helpers.get_available_capsule_port (port_pool=None)`

returns a list of unused ports dedicated for fake capsules This calls a fuser command on the server prompting for a port range. fuser commands returns a list of ports which have a PID assigned (a list of ports which are already used). This function then subtracts unavailable ports from the other ones and returns one of available ones randomly.

Parameters `port_pool` – A list of ports used for fake capsules (for RHEL7+: don't forget to set a correct selinux context before otherwise you'll get Connection Refused error)

Returns Random available port from interval <9091, 9190>.

Return type int

`robottelo.helpers.default_url_on_new_port (oldport, newport)`

Creates context where the default capsule is forwarded on a new port

Parameters

- **oldport** (*int*) – Port to be forwarded.
- **newport** (*int*) – New port to be used to forward *oldport*.

Returns A string containing the new capsule URL with port.

Return type str

class `robottelo.helpers.Storage (*args, **kwargs)`

Bases: object

Turns a dict into an attribute based object.

Example:

```
d = {'foo': 'bar'}
d['foo'] == 'bar'
storage = Storage(d)
storage.foo == 'bar'
```

`robottelo.helpers.get_func_name (func, test_item=None)`

Given a func object return standardized name to use across project

`robottelo.helpers.get_services_status ()`

Check if core services are running

`robottelo.helpers.form_repo_path (org=None, lce=None, cv=None, cvv=None, prod=None, repo=None, capsule=False)`

Forms unix path to the directory containing published repository in pulp using provided entity names. Supports both repositories in content view version and repositories in lifecycle environment. Note that either *cvv* or *lce* is required.

Parameters

- **org** (*str*) – organization label
- **optional lce** (*str*) – lifecycle environment label
- **cv** (*str*) – content view label
- **optional cvv** (*str*) – content view version, e.g. '1.0'
- **prod** (*str*) – product label
- **repo** (*str*) – repository label
- **capsule** (*bool*) – whether the `repo_path` is from a capsule or not

Returns full unix path to the specific repository

Return type str

`robottelo.helpers.create_repo` (*name*, *repo_fetch_url=None*, *packages=None*,
wipe_repodata=False, *hostname=None*)

Creates a repository from given packages and publishes it into pulp's directory for web access.

Parameters

- **name** (*str*) – repository name - name of a directory with packages
- **repo_fetch_url** (*str*) – URL to fetch packages from
- **packages** – list of packages to fetch (with extension)
- **wipe_repodata** – whether to recursively delete repodata folder
- **optional hostname** (*str*) – hostname or IP address of the remote host. If None the hostname will be get from `main.server.hostname` config.

Returns URL where the repository can be accessed

Return type str

`robottelo.helpers.repo_add_updateinfo` (*name*, *updateinfo_url=None*, *hostname=None*)

Modify repo with contents of updateinfo.xml file.

Parameters

- **name** (*str*) – repository name
- **optional updateinfo_url** (*str*) – URL to download updateinfo.xml file from. If not specified - updateinfo.xml from repository folder will be used instead
- **optional hostname** (*str*) – hostname or IP address of the remote host. If None the hostname will be get from `main.server.hostname` config.

Returns result of executing `modifyrepo` command

`robottelo.helpers.extract_capsule_satellite_installer_command` (*text*)

Extract satellite installer command from capsule-certs-generate command output

`robottelo.helpers.extract_ui_token` (*input*)

Extracts and returns the CSRF protection token from a given HTML string

`robottelo.helpers.get_web_session` ()

Logs in as admin user and returns the valid requests.Session object

`robottelo.helpers.host_provisioning_check` (*ip_addr*)

Check the provisioned host status by pinging the ip of host and check to connect to ssh port

Parameters *ip_addr* – IP address of the provisioned host

Returns ssh command return code and stdout

`robottelo.helpers.slugify_component` (*string*, *keep_hyphens=True*)

Make component name a slug

Arguments: string {*str*} – Component name e.g: ActivationKeys keep_hyphens {bool} – Keep hyphens or replace with underscores

Returns: str – component slug e.g: activationkeys

`robottelo.helpers.download_gce_cert` ()

`robottelo.helpers.idgen` (*val*)

The id generator function which will return string that will append to the parameterized test name

robottelo.host_info

Module that gather several informations about host

Module Contents**Classes**

<code>SatVersionDependentValues(*dcts, **kwargs)</code>	Class which return values depending on Satellite host version
---	---

Functions

<code>get_host_os_version()</code>	Fetches host's OS version through SSH
<code>get_host_sat_version()</code>	Fetches host's Satellite version through SSH
<code>_extract_sat_version(ssh_cmd)</code>	Extracts Satellite version if possible or 'Not Available' otherwise
<code>get_repo_files(repo_path, extension='rpm', hostname=None)</code>	Returns a list of repo files (for example rpms) in specific repository
<code>get_repomd_revision(repo_path, hostname=None)</code>	Fetches a revision of repository.
<code>get_sat_version()</code>	Try to read sat_version from envvar SAT_VERSION

robottelo.host_info.**LOGGER**

robottelo.host_info.**get_host_os_version()**

Fetches host's OS version through SSH :return: str with version

robottelo.host_info.**_SAT_6_2_VERSION_COMMAND** = `rpm -q satellite`

robottelo.host_info.**_SAT_6_1_VERSION_COMMAND** = `grep "VERSION" /usr/share/foreman/lib/satell...`

robottelo.host_info.**get_host_sat_version()**

Fetches host's Satellite version through SSH :return: Satellite version :rtype: version

robottelo.host_info.**_extract_sat_version(ssh_cmd)**

Extracts Satellite version if possible or 'Not Available' otherwise

Parameters `ssh_cmd` – str ssh command

Returns Satellite version

Return type str

robottelo.host_info.**get_repo_files(repo_path, extension='rpm', hostname=None)**

Returns a list of repo files (for example rpms) in specific repository directory.

Parameters

- **repo_path** (*str*) – unix path to the repo, e.g. `'/var/lib/pulp/fooRepo/'`
- **extension** (*str*) – extension of searched files. Defaults to `'rpm'`
- **optional hostname** (*str*) – hostname or IP address of the remote host. If None the hostname will be get from `main.server.hostname` config.

Returns list representing rpm package names

Return type list

`robottelo.host_info.get_repomd_revision(repo_path, hostname=None)`
Fetches a revision of repository.

Parameters

- **repo_path** (*str*) – unix path to the repo, e.g. `'/var/lib/pulp/fooRepo'`
- **optional hostname** (*str*) – hostname or IP address of the remote host. If `None` the hostname will be get from `main.server.hostname` config.

Returns string containing repository revision

Return type str

class `robottelo.host_info.SatVersionDependentValues(*dcts, **kwargs)`

Bases: object

Class which return values depending on Satellite host version

`__getitem__(self, item)`

Return value dependent on Satellite version :param item: str :return: respective Satellite version values

`robottelo.host_info.get_sat_version()`

Try to read `sat_version` from envvar `SAT_VERSION` if not available fallback to ssh connection to get it.

`robottelo.libvirt_discovery`

Utilities to create virtual host on libvirt with and without PXE boot

Hosts are virtual guests provisioned on a external `libvirt_host`. All guests images are stored on the `image_dir` path on external libvirt server.

Make sure to configure the `compute_resources` section on the configuration file. Also make sure that the `vlan_networking` section is properly configured.

Module Contents

Classes

<code>LibvirtGuest(cpu=1, ram=1024, boot_iso=False, extra_nic=False, libvirt_server=None, image_dir=None, mac=None, bridge=None)</code>	Manages a Libvirt guests to allow host discovery and provisioning
---	---

Functions

`__gen_mac_for_libvirt()`

`robottelo.libvirt_discovery.logger`

`robottelo.libvirt_discovery.__gen_mac_for_libvirt()`

exception `robottelo.libvirt_discovery.LibvirtGuestError`

Bases: Exception

Exception raised for failed virtual guests on external libvirt

```
class robottelo.libvirt_discovery.LibvirtGuest (cpu=1, ram=1024, boot_iso=False, extra_nic=False, libvirt_server=None, image_dir=None, mac=None, bridge=None)
```

Bases: object

Manages a Libvirt guests to allow host discovery and provisioning

It expects that Libvirt host is defined with image path. Make sure to call `destroy()` to stop and clean the image on the libvirt server, otherwise the virtual machine and its image will stay on the server consuming hardware resources.

It is possible to customize the `libvirt_host` and `image_dir` as per virtual machine basis. Just set the expected values when instantiating.

create (*self*)

Creates a virtual machine on the libvirt server using virt-install

Raises `robottelo.libvirt_discovery.LibvirtGuestError` – Whenever a virtual guest could not be executed.

destroy (*self*)

Destroys the virtual machine on the provisioning server

attach_nic (*self*)

Add a new NIC to existing host

__enter__ (*self*)

__exit__ (*self, *exc*)

robottelo.log

Utilities to help work with log files

Module Contents

Classes

<code>LogFile(remote_path, pattern=None)</code>	References a remote log file. The log file will be downloaded to allow
---	--

robottelo.log.**LOGS_DATA_DIR**

class robottelo.log.LogFile (*remote_path, pattern=None*)

Bases: object

References a remote log file. The log file will be downloaded to allow operate on it using python

filter (*self, pattern=None*)

Filter the log file using the pattern argument or object's pattern

robottelo.manifests

Manifest cloning tools..

Module Contents

Classes

<code>ManifestCloner</code> (<i>template=None, private_key=None, signing_key=None</i>)	Manifest cloning utility class.
<code>Manifest</code> (<i>content=None, org_environment_access=False, filename=None, name='default'</i>)	Class that holds the contents of a manifest with a generated filename

Functions

<code>clone</code> (<i>org_environment_access=False, name='default'</i>)	Clone the cached manifest and return a <code>Manifest</code> object.
<code>original_manifest</code> (<i>name='default'</i>)	Returns a <code>Manifest</code> object filed with the template manifest.
<code>upload_manifest_locked</code> (<i>org_id, manifest=None, interface=INTERFACE_API, timeout=None</i>)	Upload a manifest with locking, using the requested interface.

```
class robottelo.manifests.ManifestCloner (template=None, private_key=None, signing_key=None)
```

Bases: object

Manifest cloning utility class.

```
_download_manifest_info (self, name='default')
```

Download and cache the manifest information.

```
clone (self, org_environment_access=False, name='default')
```

Clones a RedHat-manifest file.

Change the consumer `uuid` and sign the new manifest with signing key. The certificate for the key must be installed on the candlepin server in order to accept uploading the cloned manifest.

Parameters

- **org_environment_access** – Whether to modify consumer content access mode to `org_environment` (Golden ticket enabled manifest).
- **name** – which manifest url to clone (named key-value pairs are defined as `fake_manifest.url` value in `robottelo.properties` (default: 'default')

Returns A file-like object (`BytesIO` on Python 3 and `StringIO` on Python 2) with the contents of the cloned manifest.

```
original (self, name='default')
```

Returns the original manifest as a file-like object.

Parameters name – A name of the manifest as defined in `robottelo.properties`

Be aware that using the original manifest and not removing it afterwards will make it impossible to import it to any other Organization.

Make sure to close the returned file-like object in order to clean up the memory used to store it.

```
robottelo.manifests._manifest_cloner
```

```
class robottelo.manifests.Manifest (content=None, org_environment_access=False, filename=None, name='default')
```

Bases: object

Class that holds the contents of a manifest with a generated filename based on `time.time`.

To ensure that the manifest content is closed use this class as a context manager with the `with` statement:

```
with Manifest() as manifest:
    # my fancy stuff
```

content

`__enter__(self)`

`__exit__(self, type, value, traceback)`

`robottelo.manifests.clone(org_environment_access=False, name='default')`

Clone the cached manifest and return a `Manifest` object.

Parameters

- **org_environment_access** – Whether to modify consumer content access mode to `org_environment` (Golden ticket enabled manifest).
- **name** – key name of the `fake_manifests.url` dict defined in `robottelo.properties`

Is highly recommended to use this with the `with` statement to make that the content of the manifest (file-like object) is closed properly:

```
with clone() as manifest:
    # my fancy stuff
```

`robottelo.manifests.original_manifest(name='default')`

Returns a `Manifest` object filed with the template manifest.

Parameters name – key name of the `fake_manifests.url` dict defined in `robottelo.properties`

Make sure to remove the manifest after its usage otherwiser the Satellite 6 server will not accept it anymore on any other organization.

Is highly recommended to use this with the `with` statement to make that the content of the manifest (file-like object) is closed properly:

```
with original_manifest() as manifest:
    # my fancy stuff
```

`robottelo.manifests.upload_manifest_locked(org_id, manifest=None, interface=INTERFACE_API, timeout=None)`

Upload a manifest with locking, using the requested interface.

Returns the upload result

Note: The manifest uploading is strictly locked only when using this function

Usage:

```
# for API interface
manifest = manifests.clone()
upload_manifest_locked(org_id, manifest, interface=INTERFACE_API)

# for CLI interface
manifest = manifests.clone()
upload_manifest_locked(org_id, manifest, interface=INTERFACE_CLI)
```

(continues on next page)

(continued from previous page)

```
# or in one line with default interface
result = upload_manifest_locked(org_id, manifests.clone())
subscription_id = result[id']
```

robottelo.products

Manage RH products repositories and custom repositories.

The main purpose of this feature is to manage product repositories especially the RH one in the context of a special distro and cdn settings.

The repository data creation became transparent when supplying only the target distro.

Example Usage:

We know that sat tool key = 'rhst'

Working with generic repos.

Generic repos has no way to guess the custom repo url in case of settings.cdn = false , that why the GenericRHRepo without custom url always return cdn repo data:

```
sat_repo = GenericRHRepository(key=PRODUCT_KEY_SAT_TOOLS)
print(sat_repo.cdn) >> True
# today the default distro is rhel7
print(sat_repo.distro) >> rhel7
print(sat_repo.data) >>
{
  'arch': 'x86_64',
  'cdn': True,
  'product': 'Red Hat Enterprise Linux Server',
  'releasever': None,
  'repository': 'Red Hat Satellite Tools 6.2 for RHEL 7 Server RPMs x86_64',
  'repository-id': 'rhel-7-server-satellite-tools-6.2-rpms',
  'repository-set': 'Red Hat Satellite Tools 6.2 (for RHEL 7 Server) (RPMs)'
}

# Generic CDN RH repository with specific distro "DISTRO_RHEL6"
sat_repo = GenericRHRepository(
    distro=DISTRO_RHEL6, key=PRODUCT_KEY_SAT_TOOLS)

print(sat_repo.distro) >> rhel6
print(sat_repo.data) >>
{
  'arch': 'x86_64',
  'cdn': True,
  'product': 'Red Hat Enterprise Linux Server',
  'releasever': None,
  'repository': 'Red Hat Satellite Tools 6.2 for RHEL 6 Server RPMs x86_64',
  'repository-id': 'rhel-6-server-satellite-tools-6.2-rpms',
  'repository-set': 'Red Hat Satellite Tools 6.2 (for RHEL 6 Server) (RPMs)'
}

# Generic RH repository with custom url
sat_repo = GenericRHRepository(
    key=PRODUCT_KEY_SAT_TOOLS, url='http://sat-tools.example.com')
```

(continues on next page)

(continued from previous page)

```

# because default settings.cdn=False and we have a custom url
print(sat_repo.cdn) >> False
print(sat_repo.distro) >> rhel7
print(sat_repo.data) >>
{'cdn': False, 'url': 'http://sat-tools.example.com'}

# Generic RH repository with custom url and force cdn
sat_repo = GenericRHRepository(
    key=PRODUCT_KEY_SAT_TOOLS,
    url='http://sat-tools.example.com',
    cdn=True
)
print(sat_repo.data) >>
{
'arch': 'x86_64',
'cdn': True,
'product': 'Red Hat Enterprise Linux Server',
'releasever': None,
'repository': 'Red Hat Satellite Tools 6.2 for RHEL 7 Server RPMs x86_64',
'repository-id': 'rhel-7-server-satellite-tools-6.2-rpms',
'repository-set': 'Red Hat Satellite Tools 6.2 (for RHEL 7 Server) (RPMs)'
}

# We have created a SatelliteToolsRepository that automatically detect it's
# custom url in settings, so there no need to explicitly initialise with
# url, simply the distro is needed (in case of specific one), otherwise
# the default distro will be used.

# SatelliteToolsRepository RH repo use settings urls and cdn
sat_repo = SatelliteToolsRepository()
print(sat_repo.cdn) >> False
print(sat_repo.distro) >> rhel7
print(sat_repo.data) >>
{
'cdn': False,
# part of the url was hidden
'url': 'XXXXXXXXXXXXXXXXXXXX/Tools_6_3_RHEL7/custom/'
      'Satellite_Tools_6_3_Composes/Satellite_Tools_x86_64/'
}

# SatelliteToolsRepository RH repo use settings urls with 'force cdn')
sat_repo = SatelliteToolsRepository(cdn=True)
print(sat_repo.cdn) >> True
print(sat_repo.distro) >> rhel7
print(sat_repo.data) >>
{
'arch': 'x86_64',
'cdn': True,
'product': 'Red Hat Enterprise Linux Server',
'releasever': None,
'repository': 'Red Hat Satellite Tools 6.2 for RHEL 7 Server RPMs x86_64',
'repository-id': 'rhel-7-server-satellite-tools-6.2-rpms',
'repository-set': 'Red Hat Satellite Tools 6.2 (for RHEL 7 Server) (RPMs)'
}

# we can also indicate the distro, the same as for the generic one the
# data will be switched for that distro

```

(continues on next page)

(continued from previous page)

```

# Working with RepositoryCollection using the default distro
repos_collection = RepositoryCollection(
    repositories=[
        RHELRepository(),
        SatelliteToolsRepository(),
        SatelliteCapsuleRepository(),
        CustomYumRepository(url=FAKE_0_YUM_REPO)
    ]
)

repos_collection.distro >> None
repos_collection.repos_data >>
[{'cdn': False,
  'url': 'http://XXXXXXXXX/RHEL-7/7.4/Server/x86_64/os/'},
 {'cdn': False,
  'url': 'http://XXXXXXXXXX/Tools_6_3_RHEL7/custom/'
  'Satellite_Tools_6_3_Composes/Satellite_Tools_x86_64/'
  },
 {'cdn': False,
  'url': 'http://XXXXXXXXXX/Satellite_6_3_RHEL7/custom/'
  'Satellite_6_3_Composes/Satellite_6_3_RHEL7'
  },
 {'cdn': False, 'url': 'http://inecas.fedorapeople.org/fakerepos/zoo/'}
]
repos_collection.need_subscription >> False

# Working with RepositoryCollection with force distro RHEL6 and force cdn
# on some repos
repos_collection = RepositoryCollection(
    distro=DISTRO_RHEL6,
    repositories=[
        SatelliteToolsRepository(cdn=True),
        SatelliteCapsuleRepository(),
        YumRepository(url=FAKE_0_YUM_REPO)
    ]
)
repos_collection.distro >> rhel6
repos_collection.repos_data >>
[
    {'arch': 'x86_64',
      'cdn': True,
      'product': 'Red Hat Enterprise Linux Server',
      'releasever': None,
      'repository': 'Red Hat Satellite Tools 6.2 for RHEL 6 Server RPMs '
                    'x86_64',
      'repository-id': 'rhel-6-server-satellite-tools-6.2-rpms',
      'repository-set': 'Red Hat Satellite Tools 6.2 (for RHEL 6 Server)
                        '(RPMs)'
    },
    {
      'arch': 'x86_64',
      'cdn': True,
      'product': 'Red Hat Satellite Capsule',
      'releasever': None,
      'repository': 'Red Hat Satellite Capsule 6.2 for RHEL 6 Server RPMs '
    }
]

```

(continues on next page)

(continued from previous page)

```

        'x86_64',
        'repository-id': 'rhel-6-server-satellite-capsule-6.2-rpms',
        'repository-set': 'Red Hat Satellite Capsule 6.2 (for RHEL 6 Server) '
                          '(RPMs)'
    },
    {'cdn': False, 'url': 'http://inecas.fedorapeople.org/fakerepos/zoo/'}
]
repos_collection.need_subscription >> True
# Note: satellite capsule repo will query the server for a distro and if
# the same distro as the sat server is used will use the settings url
# (if cdn=False) else it will use the cdn one.

# Please consult the RepositoryCollection for some usage functions
# also test usage located at:
# tests/foreman/cli/test_vm_install_products_package.py

```

Module Contents

Classes

<code>BaseRepository(url=None, distro=None, content_type=None)</code>	Base repository class for custom and RH repositories
<code>YumRepository(url=None, distro=None, content_type=None)</code>	Custom Yum repository
<code>DockerRepository(url=None, distro=None, upstream_name=None)</code>	Custom Docker repository
<code>PuppetRepository(: str, url: Optional[str] = None, distro: List[Dict[str, str]] = None, modules=None)</code>	Custom Puppet repository
<code>OSTreeRepository(url=None, distro=None, content_type=None)</code>	Custom OSTree repository
<code>GenericRHRepository(distro=None, key=None, cdn=False, url=None)</code>	Generic RH repository
<code>RHELRepository(distro=None, key=None, cdn=False, url=None)</code>	RHEL repository
<code>SatelliteToolsRepository(distro=None, key=None, cdn=False, url=None)</code>	Satellite Tools Repository
<code>SatelliteCapsuleRepository(distro=None, key=None, cdn=False, url=None)</code>	Satellite capsule repository
<code>VirtualizationAgentsRepository(distro=None, key=None, cdn=False, url=None)</code>	Virtualization Agents repository
<code>RHELCloudFormsTools(distro=None, key=None, cdn=False, url=None)</code>	Generic RH repository
<code>RHELAnsibleEngineRepository(distro=None, key=None, cdn=False, url=None)</code>	Red Hat Ansible Engine Repository
<code>RepositoryCollection(distro=None, repositories=None)</code>	Repository collection

Functions

`get_server_distro()`

`robottelo.products.REPO_TYPE_YUM`

`robottelo.products.REPO_TYPE_DOCKER`

`robottelo.products.REPO_TYPE_PUPPET`

`robottelo.products.REPO_TYPE_OSTREE`

`robottelo.products.DOWNLOAD_POLICY_ON_DEMAND = on_demand`

`robottelo.products.DOWNLOAD_POLICY_IMMEDIATE = immediate`

`robottelo.products.DOWNLOAD_POLICY_BACKGROUND = background`

`robottelo.products.PRODUCT_KEY_RHEL = rhel`

`robottelo.products.PRODUCT_KEY_SAT_TOOLS = rhst`

`robottelo.products.PRODUCT_KEY_SAT_CAPSULE = rhsc`

`robottelo.products.PRODUCT_KEY_VIRT_AGENTS = rhva6`

`robottelo.products.PRODUCT_KEY_CLOUD_FORMS_TOOLS = rhct6`

`robottelo.products.PRODUCT_KEY_ANSIBLE_ENGINE = rhae2`

`robottelo.products._server_distro :str`

exception `robottelo.products.RepositoryAlreadyDefinedError`
Bases: Exception

Raised when a repository has already a predefined key

exception `robottelo.products.DistroNotSupportedError`
Bases: Exception

Raised when using a non supported distro

exception `robottelo.products.RepositoryDataNotFound`
Bases: Exception

Raised when repository data cannot be found for a predefined distro

exception `robottelo.products.OnlyOneOSRepositoryAllowed`
Bases: Exception

Raised when trying to more than one OS repository to a collection

exception `robottelo.products.RepositoryAlreadyCreated`
Bases: Exception

Raised when a repository content is already created and trying to launch the create an other time

exception `robottelo.products.ReposContentSetupWasNotPerformed`
Bases: Exception

Raised when trying to setup a VM but the repositories content was not setup

`robottelo.products.get_server_distro () → str`

class `robottelo.products.BaseRepository (url=None, distro=None, content_type=None)`
Bases: object

Base repository class for custom and RH repositories

```

    _url :Optional[str]
    _distro :Optional[str]
    _type :Optional[str]
    _repo_info :Optional[Dict]
    url :Optional[str]
    cdn :bool
    data :Dict
    distro :Optional[str]
        Return the current distro
    content_type :str
    repo_info :Optional[Dict]
    __repr__(self)
        Return repr(self).
    create(self: int, organization_id: int, product_id: str, download_policy: bool = DOWN-
        LOAD_POLICY_ON_DEMAND, synchronize=True)
        Create the repository for the supplied product id
    synchronize(self)
        Synchronize the repository
    add_to_content_view(self: int, organization_id: int, content_view_id)
        Associate repository content to content-view
class robottelo.products.YumRepository(url=None, distro=None, content_type=None)
    Bases: robottelo.products.BaseRepository
    Custom Yum repository
    _type :str
class robottelo.products.DockerRepository(url=None, distro=None, up-
    stream_name=None)
    Bases: robottelo.products.BaseRepository
    Custom Docker repository
    _type :str
    upstream_name
    create(self, organization_id, product_id, download_policy=None, synchronize=True)
        Create the repository for the supplied product id
class robottelo.products.PuppetRepository(: str, url: Optional[str] = None, distro:
    List[Dict[str, str]] = None, modules=None)
    Bases: robottelo.products.BaseRepository
    Custom Puppet repository
    _type :str
    puppet_modules
    add_to_content_view(self: int, organization_id: int, content_view_id)
        Associate repository content to content-view

```

```
class robottelo.products.OSTreeRepository (url=None, distro=None, content_type=None)
```

```
    Bases: robottelo.products.BaseRepository
```

```
    Custom OSTree repository
```

```
    _type
```

```
class robottelo.products.GenericRHRepository (distro=None, key=None, cdn=False,  
                                              url=None)
```

```
    Bases: robottelo.products.BaseRepository
```

```
    Generic RH repository
```

```
    _type
```

```
    _distro :str
```

```
    _key :str
```

```
    _repo_data :Dict
```

```
    _url :Optional[str]
```

```
    url
```

```
    cdn
```

```
    key
```

```
    distro
```

```
        Return the current distro
```

```
    repo_data :Dict
```

```
    is_distro_repository :bool
```

```
    distro_major_version
```

```
    distro_repository :Optional[RHELRepository]
```

```
        Return the OS distro repository object relied to this repository
```

```
        Suppose we have a repository for a product that must be installed on RHEL, but for proper installation needs some dependencies packages from the OS repository. This function will return the right OS repository object for later setup.
```

```
        for example: capsule_repo = SatelliteCapsuleRepository() # the capsule_repo will represent a capsule repo for default distro  
                    rhel_repo = capsule_repo.distro_repository # the rhel repo representation object for default distro will be # returned, if not found raise exception
```

```
    rh_repository_id :Optional[str]
```

```
    data :Dict
```

```
    _get_repo_data (self: Optional[str], distro=None)
```

```
        Return the repo data as registered in constant module and bound to distro.
```

```
    _repo_is_distro (self: Optional[Dict], repo_data=None)
```

```
        return whether the repo data is for an OS distro product repository
```

```
    __repr__ (self)
```

```
        Return repr(self).
```

```
    create (self: int, organization_id: Optional[int], product_id: Optional[str] = None, download_policy: Optional[bool] = DOWNLOAD_POLICY_ON_DEMAND, synchronize=True)
```

```
        Create an RH repository
```

```

class robottelo.products.RHELRepository (distro=None, key=None, cdn=False, url=None)
    Bases: robottelo.products.GenericRHRepository
    RHEL repository
    _key
    url

class robottelo.products.SatelliteToolsRepository (distro=None, key=None, cdn=False, url=None)
    Bases: robottelo.products.GenericRHRepository
    Satellite Tools Repository
    _key
    url

class robottelo.products.SatelliteCapsuleRepository (distro=None, key=None, cdn=False, url=None)
    Bases: robottelo.products.GenericRHRepository
    Satellite capsule repository
    _key
    url

class robottelo.products.VirtualizationAgentsRepository (distro=None, key=None, cdn=False, url=None)
    Bases: robottelo.products.GenericRHRepository
    Virtualization Agents repository
    _key
    _distro

class robottelo.products.RHELCloudFormsTools (distro=None, key=None, cdn=False, url=None)
    Bases: robottelo.products.GenericRHRepository
    Generic RH repository
    _distro
    _key

class robottelo.products.RHELAnsibleEngineRepository (distro=None, key=None, cdn=False, url=None)
    Bases: robottelo.products.GenericRHRepository
    Red Hat Ansible Engine Repository
    _key

class robottelo.products.RepositoryCollection (distro=None, repositories=None)
    Bases: object
    Repository collection
    _distro :str
    _org :Dict
    _items :List[BaseRepository] = []
    _repos_info :List[Dict] = []

```

```
_custom_product_info :Dict
_os_repo :RHELRepository
_setup_content_data :Dict[str, Dict]
distro :str
repos_info :List[Dict]
custom_product
os_repo :RHELRepository
repos_data :List[Dict]
rh_repos :List[BaseRepository]
custom_repos :List[BaseRepository]
rh_repos_info :List[Dict]
custom_repos_info :List[Dict]
setup_content_data
need_subscription :bool
organization
add_item(self, item)
    Add repository to collection
    Parameters item (BaseRepository) – Item to add
    Returns None
add_items(self, items)
    Add multiple repositories to collection
    Parameters items (List [BaseRepository]) – Items to add
    Returns None
__iter__(self)
setup(self: int, org_id: str, download_policy: bool = DOWNLOAD_POLICY_ON_DEMAND, synchro-
    nize=True)
    Setup the repositories on server.
    Recommended usage: repository only setup, for full content setup see setup_content.
setup_content_view(self: int, org_id: int, lce_id=None)
    Setup organization content view by adding all the repositories, publishing and promoting to lce if needed.
static setup_activation_key(org_id: int, content_view_id: int, lce_id: int, subscrip-
    tion_names: Optional[List[str]] = None)
    Create activation and associate content-view, lifecycle environment and subscriptions
static organization_has_manifest(organization_id)
    Check if an organization has a manifest, an organization has manifest if one of it's subscriptions have the
    account defined.
setup_content(self: int, org_id: int, lce_id: bool, upload_manifest: str = False,
    download_policy: Optional[List[str]] = DOWNLOAD_POLICY_ON_DEMAND,
    rh_subscriptions=None)
    Setup content view and activation key of all the repositories.
```

Parameters

- **org_id** – The organization id
- **lce_id** – The lifecycle environment id
- **upload_manifest** – Whether to upload the manifest (The manifest is uploaded only if needed)
- **download_policy** – The repositories download policy
- **rh_subscriptions** – The RH subscriptions to be added to activation key

setup_virtual_machine (*self*, *vm*, *patch_os_release=False*, *install_katello_agent=True*, *enable_rh_repos=True*, *enable_custom_repos=False*, *configure_rhel_repo=False*)

Setup The virtual machine basic task, eg: install katello ca, register vm host, enable rh repos and install katello-agent

Parameters

- **vm** (*VirtualMachine*) – The Virtual machine to setup.
- **patch_os_release** (*bool*) – whether to patch the VM with os version.
- **install_katello_agent** (*bool*) – whether to install katello-agent
- **enable_rh_repos** (*bool*) – whether to enable RH repositories
- **enable_custom_repos** (*bool*) – whether to enable custom repositories
- **configure_rhel_repo** (*bool*) – Whether to configure the distro Red Hat repository, this is needed to configure manually RHEL custom repo url as sync time is very big (more than 2 hours for RHEL 7Server) and not critical for some contexts.

robottelo.rhssso_utils

Utility module to handle the rhssso-satellite configure UI/CLI/API testing

Module Contents

Functions

<code>run_command(cmd, hostname=satellite, timeout=None)</code>	helper function for ssh command and avoiding the return code check in called function
<code>get_rhssso_client_id()</code>	Getter method for fetching the client id and can be used other functions
<code>get_rhssso_user_details(username)</code>	Getter method to receive the user id
<code>upload_rhssso_entity(json_content, entity_name)</code>	Helper method upload the entity json request as file on RHSSO Server
<code>create_mapper(json_content, client_id)</code>	Helper method to create the RH-SSO Client Mapper
<code>create_new_rhssso_user(client_id, user_name=None)</code>	create new user in RHSSO instance and set the password
<code>delete_rhssso_user(username)</code>	Delete the RHSSO user

robottelo.rhssso_utils.**satellite**

robottelo.rhssso_utils.**rhssso_host**

```
robottelo.rhssso_utils.realm
robottelo.rhssso_utils.rhssso_user
robottelo.rhssso_utils.rhssso_password
robottelo.rhssso_utils.run_command(cmd, hostname=satellite, timeout=None)
    helper function for ssh command and avoiding the return code check in called function
robottelo.rhssso_utils.get_rhssso_client_id()
    Getter method for fetching the client id and can be used other functions
robottelo.rhssso_utils.get_rhssso_user_details(username)
    Getter method to receive the user id
robottelo.rhssso_utils.upload_rhssso_entity(json_content, entity_name)
    Helper method upload the entity json request as file on RHSSO Server
robottelo.rhssso_utils.create_mapper(json_content, client_id)
    Helper method to create the RH-SSO Client Mapper
robottelo.rhssso_utils.create_new_rhssso_user(client_id, username=None)
    create new user in RHSSO instance and set the password
robottelo.rhssso_utils.delete_rhssso_user(username)
    Delete the RHSSO user
```

robottelo.ssh

Utility module to handle the shared ssh connection.

Module Contents

Classes

<code>SSHCommandResult(stdout=None, stderr=None, return_code=0, output_format=None)</code>	Structure that returns in all ssh commands results.
<code>SSHClient()</code>	Extended SSHClient allowing custom methods

Functions

<code>decode_to_utf8(text)</code>	Paramiko returns bytes object and we need to ensure it is utf-8 before
<code>_call_paramiko_sshclient()</code>	Call <code>paramiko.SSHClient</code> .
<code>get_client(hostname=None, username=None, password=None, key_filename=None, timeout=None, port=22)</code>	Returns a SSH client connected to given hostname
<code>get_connection(hostname=None, username=None, password=None, key_filename=None, timeout=None, port=22)</code>	Yield an ssh connection object.
<code>get_sftp_session(hostname=None, username=None, password=None, key_filename=None, timeout=None)</code>	Yield a SFTP session object.

Continued on next page

Table 107 – continued from previous page

<code>add_authorized_key(key, hostname=None, username=None, password=None, key_filename=None, timeout=None)</code>	Appends a local public ssh key to remote authorized keys
<code>upload_file(local_file, remote_file, key_filename=None, hostname=None)</code>	Upload a local file to a remote machine
<code>upload_files(local_dir, remote_dir, file_search='*.txt', hostname=None, key_filename=None)</code>	Upload all files from directory to a remote directory
<code>_upload_file(sftp, local_file, remote_file)</code>	Upload a file using existent sftp session
<code>download_file(remote_file, local_file=None, hostname=None)</code>	Download a remote file to the local machine. If hostname is not
<code>command(cmd, hostname=None, output_format=None, username=None, password=None, key_filename=None, timeout=None, connection_timeout=None, port=22)</code>	Executes SSH command(s) on remote hostname.
<code>execute_command(cmd, connection, output_format=None, timeout=None, connection_timeout=None)</code>	Execute a command via ssh in the given connection
<code>is_ssh_pub_key(key)</code>	Validates if a string is in valid ssh pub key format

`robottelo.ssh.logger`

exception `robottelo.ssh.SSHCommandTimeoutError`

Bases: `Exception`

Raised when the SSH command has not finished executing after a predefined period of time.

`robottelo.ssh.decode_to_utf8(text)`

Paramiko returns bytes object and we need to ensure it is utf-8 before parsing

class `robottelo.ssh.SSHCommandResult(stdout=None, stderr=None, return_code=0, output_format=None)`

Bases: `object`

Structure that returns in all ssh commands results.

`__repr__(self)`

Return `repr(self)`.

class `robottelo.ssh.SSHClient`

Bases: `paramiko.SSHClient`

Extended `SSHClient` allowing custom methods

run (*self*, *cmd*, *args, **kwargs)

This method exists to allow the reuse of existing connections when running multiple ssh commands as in the following example of use:

```
with robotello.ssh.get_connection() as connection:
    connection.run('ls /tmp')
    connection.run('another command')
```

self is always passed as the connection when used in context manager only when using `ssh.get_connection` function.

Note: This method is named `run` to avoid conflicts with existing `exec_command` and local function `execute_command`.

`robottelo.ssh._call_paramiko_sshclient()`

Call `paramiko.SSHClient`.

This function does not alter the behaviour of `paramiko.SSHClient`. It exists solely for the sake of easing unit testing: it can be overridden for mocking purposes.

`robottelo.ssh.get_client` (*hostname=None, username=None, password=None, key_filename=None, timeout=None, port=22*)

Returns a SSH client connected to given hostname

`robottelo.ssh.get_connection` (*hostname=None, username=None, password=None, key_filename=None, timeout=None, port=22*)

Yield an ssh connection object.

The connection will be configured with the specified arguments or will fall-back to server configuration in the configuration file.

Yield this SSH connection. The connection is automatically closed when the caller is done using it using `contextlib`, so clients should use the `with` statement to handle the object:

```
with get_connection() as connection:
    ...
```

Parameters

- **hostname** (*str*) – The hostname of the server to establish connection. If it is `None` hostname from configuration's `server` section will be used.
- **username** (*str*) – The username to use when connecting. If it is `None` `ssh_username` from configuration's `server` section will be used.
- **password** (*str*) – The password to use when connecting. If it is `None` `ssh_password` from configuration's `server` section will be used. Should be applied only in case `key_filename` is not set
- **key_filename** (*str*) – The path of the ssh private key to use when connecting to the server. If it is `None` `key_filename` from configuration's `server` section will be used.
- **timeout** (*int*) – Time to wait for establish the connection.
- **port** (*int*) – The server port to connect to, the default port is 22.

Returns An SSH connection.

Return type `paramiko.SSHClient`

`robottelo.ssh.get_sftp_session` (*hostname=None, username=None, password=None, key_filename=None, timeout=None*)

Yield a SFTP session object.

The session will be configured with the host whose hostname is passed as argument.

Yield this SFTP Session. The session is automatically closed when the caller is done using it using `contextlib`, so clients should use the “with” statement to handle the object:

```
with get_sftp_session() as session:
    ...
```

Parameters

- **hostname** (*str*) – The hostname of the server to establish connection. If it is `None` hostname from configuration's `server` section will be used.
- **username** (*str*) – The username to use when connecting. If it is `None` `ssh_username` from configuration's `server` section will be used.

- **password** (*str*) – The password to use when connecting. If it is `None` `ssh_password` from configuration's `server` section will be used. Should be applied only in case `key_filename` is not set
- **key_filename** (*str*) – The path of the ssh private key to use when connecting to the server. If it is `None` `key_filename` from configuration's `server` section will be used.
- **timeout** (*int*) – Time to wait for establish the connection.

`robottelo.ssh.add_authorized_key` (*key*, *hostname=None*, *username=None*, *password=None*, *key_filename=None*, *timeout=None*)

Appends a local public ssh key to remote authorized keys

refer to: `remote_execution_ssh_keys` provisioning template

Parameters

- **key** – either a file path, key string or a file-like obj to append.
- **hostname** (*str*) – The hostname of the server to establish connection. If it is `None` `hostname` from configuration's `server` section will be used.
- **username** (*str*) – The username to use when connecting. If it is `None` `ssh_username` from configuration's `server` section will be used.
- **password** (*str*) – The password to use when connecting. If it is `None` `ssh_password` from configuration's `server` section will be used. Should be applied only in case `key_filename` is not set
- **key_filename** (*str*) – The path of the ssh private key to use when connecting to the server. If it is `None` `key_filename` from configuration's `server` section will be used.
- **timeout** (*int*) – Time to wait for establish the connection.

`robottelo.ssh.upload_file` (*local_file*, *remote_file*, *key_filename=None*, *hostname=None*)

Upload a local file to a remote machine

Parameters

- **local_file** – either a file path or a file-like object to be uploaded.
- **remote_file** – a remote file path where the uploaded file will be placed.
- **hostname** – target machine hostname. If not provided will be used the `server.hostname` from the configuration.
- **key_filename** (*str*) – The path of the ssh private key to use when connecting to the server. If it is `None` `key_filename` from configuration's `server` section will be used.

`robottelo.ssh.upload_files` (*local_dir*, *remote_dir*, *file_search='*.txt'*, *hostname=None*, *key_filename=None*)

Upload all files from directory to a remote directory

Parameters

- **local_dir** – all files from local path to be uploaded.
- **remote_dir** – a remote path where the uploaded files will be placed.
- **file_search** – filter only files contains the type extension
- **hostname** – target machine hostname. If not provided will be used the `server.hostname` from the configuration.
- **key_filename** (*str*) – The path of the ssh private key to use when connecting to the server. If it is `None` `key_filename` from configuration's `server` section will be used.

`robottelo.ssh._upload_file` (*sftp, local_file, remote_file*)

Upload a file using existent sftp session

Parameters

- **sftp** – sftp session object
- **local_file** – either a file path or a file-like object to be uploaded.
- **remote_file** – a remote file path where the uploaded file will be placed.

`robottelo.ssh.download_file` (*remote_file, local_file=None, hostname=None*)

Download a remote file to the local machine. If `hostname` is not provided will be used the server.

`robottelo.ssh.command` (*cmd, hostname=None, output_format=None, username=None, password=None, key_filename=None, timeout=None, connection_timeout=None, port=22*)

Executes SSH command(s) on remote hostname.

Parameters

- **cmd** (*str*) – The command to run
- **output_format** (*str*) – json, csv or None
- **hostname** (*str*) – The hostname of the server to establish connection. If it is None `hostname` from configuration's `server` section will be used.
- **username** (*str*) – The username to use when connecting. If it is None `ssh_username` from configuration's `server` section will be used.
- **password** (*str*) – The password to use when connecting. If it is None `ssh_password` from configuration's `server` section will be used. Should be applied only in case `key_filename` is not set
- **key_filename** (*str*) – The path of the ssh private key to use when connecting to the server. If it is None `key_filename` from configuration's `server` section will be used.
- **timeout** (*int*) – Time to wait for the ssh command to finish.
- **connection_timeout** – Time to wait for establishing the connection.
- **port** (*int*) – The server port to connect to, the default port is 22.

`robottelo.ssh.execute_command` (*cmd, connection, output_format=None, timeout=None, connection_timeout=None*)

Execute a command via ssh in the given connection

Parameters

- **cmd** – a command to be executed via ssh
- **connection** – SSH Paramiko client connection
- **output_format** – baseljsonlcsvllist valid only for hammer commands
- **timeout** – Time to wait for the ssh command to finish.
- **connection_timeout** – Time to wait for establishing the connection.

Returns SSHCommandResult

`robottelo.ssh.is_ssh_pub_key` (*key*)

Validates if a string is in valid ssh pub key format

Parameters **key** – A string containing a ssh public key encoded in base64

Returns Boolean

robottelo.system_facts

JSON representation for a RHEL server.

Module Contents

Functions

<code>__bios_date()</code>	Generate a random date for system's BIOS between
<code>generate_system_facts(name=None)</code>	Generate random system facts for registration.

`robottelo.system_facts.__bios_date()`

Generate a random date for system's BIOS between today and 10 years ago.

Returns A random *datetime.date* that falls within the last 10 years from today.

Return type object

`robottelo.system_facts.ARCHITECTURES = ['i386', 'x86_64', 'ppc', 's390x']`

`robottelo.system_facts.DISTRO_IDS`

`robottelo.system_facts.MEMORY_CAPACITY = ['2 GB', '4 GB', '8 GB', '16 GB']`

`robottelo.system_facts.MEMORY_SIZE = ['1024 MB', '2048 MB', '4096 MB', '8192 MB']`

`robottelo.system_facts.SYSTEM_FACTS`

`robottelo.system_facts.generate_system_facts(name=None)`

Generate random system facts for registration.

Parameters `name` (*str*) – A valid FQDN for a system. If one is not provided, then a random value will be generated.

Returns A dictionary with random system facts

Return type dict

robottelo.test

Test utilities for writing foreman tests

All test cases for foreman tests are defined in this module and have utilities to help writing API and CLI tests.

Module Contents

Classes

<code>NotRaisesValueHandler()</code>	Base class for handling exception values for AssertNotRaises. Child
<code>APINotRaisesValueHandler(expected_value)</code>	AssertNotRaises value handler for API status code.
<code>CLINotRaisesValueHandler(expected_value)</code>	AssertNotRaises value handler for CLI return code.

Continued on next page

Table 109 – continued from previous page

<code>__AssertNotRaisesContext</code> (expected, value_handler_class, failure_exception, expected_regex=None, value_handler=None, expected_value=None)	A context manager used to implement <code>TestCase.assertNotRaises()</code> .
<code>AssertCliNotRaisesContextManager</code> (expected, expected_regex=None, value_handler=None, expected_value=None)	A context manager used to implement <code>TestCase.assertNotRaises()</code> .
<code>AssertApiNotRaisesContextManager</code> (expected, expected_regex=None, value_handler=None, expected_value=None)	A context manager used to implement <code>TestCase.assertNotRaises()</code> .
<code>TestCase()</code>	Robottelo test case
<code>APITestCase()</code>	Test case for API tests.
<code>CLITestCase()</code>	Test case for CLI tests.

robottelo.test.**LOGGER**

class robottelo.test.**NotRaisesValueHandler**

Bases: object

Base class for handling exception values for AssertNotRaises. Child classes can be used to validate whether specific for interface expected value is present in exception.

value_name

Property used to return expected value name (e.g. 'status code' or 'return code').

validate (*self*, *exception*)

Validate whether expected value is present in exception.

class robottelo.test.**APINotRaisesValueHandler** (*expected_value*)

Bases: `robottelo.test.NotRaisesValueHandler`

AssertNotRaises value handler for API status code.

value_name

Returns API expected value name (status code)

validate (*self*, *exception*)

Validate whether expected status code is present in specific exception.

class robottelo.test.**CLINotRaisesValueHandler** (*expected_value*)

Bases: `robottelo.test.NotRaisesValueHandler`

AssertNotRaises value handler for CLI return code.

value_name

Returns CLI expected value name (return code)

validate (*self*, *exception*)

Validate whether expected return code is present in specific exception.

class robottelo.test.**__AssertNotRaisesContext** (*expected*, *value_handler_class*, *failure_exception*, *expected_regex=None*, *expected_value=None*, *value_handler=None*)

Bases: object

A context manager used to implement `TestCase.assertNotRaises()`.

__enter__ (*self*)

__exit__ (*self*, *exc_type*, *exc_value*, *tb*)

class `robottelo.test.AssertCliNotRaisesContextManager` (*expected*, *ex-*
pected_regex=None,
expected_value=None,
value_handler=None)

Bases: `robottelo.test._AssertNotRaisesContext`

A context manager used to implement `TestCase.assertNotRaises()`.

class `robottelo.test.AssertApiNotRaisesContextManager` (*expected*, *ex-*
pected_regex=None,
expected_value=None,
value_handler=None)

Bases: `robottelo.test._AssertNotRaisesContext`

A context manager used to implement `TestCase.assertNotRaises()`.

class `robottelo.test.TestCase`

Bases: `unittest2.TestCase`

Robottelo test case

`_default_interface`

`_default_notraises_value_handler`

classmethod `setUpClass` (*cls*)

classmethod `tearDownClass` (*cls*)

classmethod `upload_manifest` (*cls*, *org_id*, *manifest*, *interface=None*, *timeout=None*)

Upload manifest locked using the default TestCase manifest if interface not specified.

Usage:

```
manifest = manifests.clone()
self.upload_manifest(org_id, manifest)

# or if you want to specify explicitly an interface
manifest = manifests.clone()
self.upload_manifest(org_id, manifest, interface=INTERFACE_CLI)

# in one line
result = self.upload_manifest(org_id, manifests.clone())
```

assertNotRaises (*self*, *expected_exception*, *callableObj=None*, *expected_value=None*,
value_handler=None, **args*, ***kwargs*)

Fail if an exception of class `expected_exception` is raised by `callableObj` when invoked with specified positional and keyword arguments. If a different type of exception is raised, it will not be caught, and the test case will be deemed to have suffered an error, exactly as for an unexpected exception.

If called with `callableObj` omitted or `None`, will return a context object used like this:

```
with self.assertNotRaises(SomeException):
    do_something()
```

The context manager keeps a reference to the exception as the `'exception'` attribute. This allows you to inspect the exception after the assertion:

```
with self.assertNotRaises(SomeException) as cm:
    do_something()
the_exception = cm.exception
self.assertEqual(the_exception.error_code, 1)
```

In addition, optional 'http_status_code' or 'cli_return_code' arg may be passed. This allows to specify exact HTTP status code or CLI return code, returned by `requests.HTTPError` or `robottelo.cli.base.CLIReturnCodeError` accordingly, which should be validated. In such case only expected exception with expected response code will be caught.

assertNotRaisesRegex (*self*, *expected_exception*, *expected_regex*, *callableObj=None*, *expected_value=None*, *value_handler=None*, **args*, ***kwargs*)

Fail if an exception of class `expected_exception` is raised and the message in the exception matches a regex.

class `robottelo.test.APITestCase`

Bases: `robottelo.test.TestCase`

Test case for API tests.

`_default_interface`

`_default_notraises_value_handler`

`_multiprocess_can_split_ = True`

class `robottelo.test.CLITestCase`

Bases: `robottelo.test.TestCase`

Test case for CLI tests.

`_default_interface`

`_default_notraises_value_handler`

`_multiprocess_can_split_ = True`

assert_error_msg (*self*, *raise_ctx*, **contents*)

Checking error msg present on Raise Context Exception Raise assertion error if any of contents are not present on error msg

Parameters

- **raise_ctx** – Raise Context
- **contents** – contents which must be present on message

robottelo.upgrade_utility

Common Upgrade test utilities

Module Contents

Functions

<code>run_goferd(client_hostname=None)</code>	Start the goferd process.
<code>check_package_installed(client_hostname=None, package=None)</code>	Verify if package is installed on docker content host.
<code>install_or_update_package(client_hostname=None, update=False, package=None)</code>	Install/update the package on docker content host.
<code>create_repo(rpm_name, post_upgrade=False, other_rpm=None)</code>	Creates a custom yum repository, that will be synced to satellite
<code>host_status(client_container_name=None)</code>	fetch the content host details.

Continued on next page

Table 110 – continued from previous page

<code>host_location_update</code> (<code>client_container_name=None</code> , Check the content host status (as package profile update <code>logger_obj=None</code> , <code>loc=None</code>) task does
<code>publish_content_view</code> (<code>org=None</code> , <code>repolist=None</code>) publish content view and return content view

`robottelo.upgrade_utility.run_goferd` (`client_hostname=None`)

Start the goferd process. :param: str client_hostname: It should be container's id.

`robottelo.upgrade_utility.check_package_installed` (`client_hostname=None`, `package=None`)

Verify if package is installed on docker content host. :param: str client_hostname: It should be container's id.
:param: str package: pass the package name to check the status :return: name of the installed package

`robottelo.upgrade_utility.install_or_update_package` (`client_hostname=None`, `update=False`, `package=None`)

Install/update the package on docker content host. :param: str client_hostname: It should be container's id.
:param: bool update: :param: str package:

`robottelo.upgrade_utility.create_repo` (`rpm_name`, `repo_path`, `post_upgrade=False`, `other_rpm=None`)

Creates a custom yum repository, that will be synced to satellite and later to capsule from satellite :param: str
rpm_name : rpm name, required to create a repository. :param: str repo_path: Name of the repository path
:param: bool post_upgrade: For Pre-upgrade, post_upgrade value will be False :param: str other_rpm: If we
want to clean a specific rpm and update with latest then we pass other rpm.

`robottelo.upgrade_utility.host_status` (`client_container_name=None`)

fetch the content host details. :param: str client_container_name: The content host hostname :return: nail-
gun.entity.host: host

`robottelo.upgrade_utility.host_location_update` (`client_container_name=None`, `logger_obj=None`, `loc=None`)

Check the content host status (as package profile update task does take time to upload) and update location.

Param str client_container_name: The content host hostname

Param str loc: Location

`robottelo.upgrade_utility.publish_content_view` (`org=None`, `repolist=None`)

publish content view and return content view :param: str org: Name of the organisation :param: str repolist:
Name of the repolist :return: Return content view

robottelo.virtwho_utils

Utility module to handle the virtwho configure UI/CLI/API testing

Module Contents

Functions

<code>_parse_entry</code> (<code>entry</code>)	Parse the string and return json format
<code>get_system</code> (<code>system_type</code>)	Return a dict account for ssh connect.
<code>get_guest_info</code> ()	Return the <code>guest_name</code> , <code>guest_uuid</code>
<code>runcmd</code> (<code>cmd</code> , <code>system=None</code> , <code>timeout=600</code> , <code>out-put_format='base'</code>)	Return the retcode and stdout.

Continued on next page

Table 111 – continued from previous page

<code>register_system(system, activation_key=None, org='Default_Organization', env='Library')</code>	Return True if the system is registered to satellite successfully.
<code>virtwho_cleanup()</code>	Before running test cases, need to clean the environment.
<code>get_virtwho_status()</code>	Return the status of virt-who service, it will help us to know
<code>get_configure_id(name)</code>	Return the configure id by hammer.
<code>get_configure_command(config_id, org=DEFAULT_ORG)</code>	Return the deploy command line based on configure id.
<code>get_configure_file(config_id)</code>	Return the configuration file full name in /etc/virt-who.d
<code>get_configure_option(option, filename)</code>	Return the option's value for the specific file.
<code>_get_hypervisor_mapping(logs)</code>	Analysing rhsm.log and get to know: what is the hypervisor_name
<code>deploy_validation()</code>	Checkout the deploy result
<code>deploy_configure_by_command(command, debug=False, org='Default_Organization')</code>	Deploy and run virt-who service by the hammer command.
<code>deploy_configure_by_script(script_content, debug=False)</code>	Deploy and run virt-who service by the shell script.
<code>restart_virtwho_service()</code>	Do the following:
<code>update_configure_option(option, value, config_file)</code>	Update option in virt-who config file
<code>delete_configure_option(option, config_file)</code>	Delete option in virt-who config file
<code>add_configure_option(option, value, config_file)</code>	Add option to virt-who config file
<code>hypervisor_json_create(hypervisors, guests)</code>	Create a hypervisor guest json data. For example:
<code>create_fake_hypervisor_content(org_label, hypervisors, guests)</code>	Post the fake hypervisor content to satellite server
<code>get_hypervisor_info()</code>	Get the hypervisor_name and guest_name from rhsm.log.
<code>virtwho_package_locked()</code>	Uninstall virt-who package and lock the foreman-maintain packages.

`robottelo.virtwho_utils.VIRTWHO_SYSCONFIG = /etc/sysconfig/virt-who`

exception `robottelo.virtwho_utils.VirtWhoError`

Bases: Exception

Exception raised for failed virtwho operations

`robottelo.virtwho_utils._parse_entry(entry)`

Parse the string and return json format

`robottelo.virtwho_utils.get_system(system_type)`

Return a dict account for ssh connect.

Parameters `system_type` (*str*) – The type of the system, should be one of ('satellite', 'guest').

Raises `VirtWhoError`: If wrong `system_type` specified.

`robottelo.virtwho_utils.get_guest_info()`

Return the guest_name, guest_uuid

`robottelo.virtwho_utils.runcmd(cmd, system=None, timeout=600, output_format='base')`

Return the retcode and stdout.

Parameters

- **cmd** (*str*) – The command line will be executed in the target system.
- **system** (*dict*) – the system account which ssh will connect to, it will connect to the satellite host if the system is None.

- **timeout** (*int*) – Time to wait for establish the connection.
- **output_format** (*str*) – baseljsonlcsvllist

`robottelo.virtwho_utils.register_system(system, activation_key=None, org='Default_Organization', env='Library')`

Return True if the system is registered to satellite successfully.

Parameters

- **system** (*dict*) – system account used by ssh to connect and register.
- **activation_key** (*str*) – the activation key will be used to register.
- **org** (*str*) – Which organization will be used to register.
- **env** (*str*) – Which environment will be used to register.

Raises `VirtWhoError`: If failed to register the system.

`robottelo.virtwho_utils.virtwho_cleanup()`

Before running test cases, need to clean the environment. Do the following: 1. stop virt-who service. 2. kill all the virt-who pid 3. clean rhsm.log message, make sure there is no old message exist. 4. clean all the configure files in `/etc/virt-who.d/`

`robottelo.virtwho_utils.get_virtwho_status()`

Return the status of virt-who service, it will help us to know the virt-who configuration file is deployed or not.

`robottelo.virtwho_utils.get_configure_id(name)`

Return the configure id by hammer. :param str name: the configure name you have created. :raises: `VirtWhoError`: If failed to get the configure info by hammer.

`robottelo.virtwho_utils.get_configure_command(config_id, org=DEFAULT_ORG)`

Return the deploy command line based on configure id. :param str config_id: the unique id of the configure file you have created. :param str org: the satellite organization name.

`robottelo.virtwho_utils.get_configure_file(config_id)`

Return the configuration file full name in `/etc/virt-who.d` :param str config_id: the unique id of the configuration file you have created.

`robottelo.virtwho_utils.get_configure_option(option, filename)`

Return the option's value for the specific file.

Parameters

- **option** (*str*) – the option name in the configuration file
- **filename** (*str*) – the configuration file, it could be: `/etc/sysconfig/virt-who` `/etc/virt-who.d/virt-who-config-{}.conf`

Raises `VirtWhoError`: If this option name not in the file.

`robottelo.virtwho_utils._get_hypervisor_mapping(logs)`

Analysing rhsm.log and get to know: what is the hypervisor_name for the specific guest. :param str logs: the output of rhsm.log. :raises: `VirtWhoError`: If hypervisor_name is None. :return: hypervisor_name and guest_name

`robottelo.virtwho_utils.deploy_validation()`

Checkout the deploy result :raises: `VirtWhoError`: If failed to start virt-who servcie. :return: hypervisor_name and guest_name

`robottelo.virtwho_utils.deploy_configure_by_command(command, debug=False, org='Default_Organization')`

Deploy and run virt-who servcie by the hammer command.

Parameters

- **command** (*str*) – get the command by UI/CLI/API, it should be like: *hammer virt-who-config deploy -id 1 -organization-id 1*
- **debug** (*bool*) – if VIRTWHO_DEBUG=1, this option should be True.

`robottelo.virtwho_utils.deploy_configure_by_script` (*script_content, debug=False*)

Deploy and run virt-who service by the shell script. :param str script_content: get the script by UI or API. :param bool debug: if VIRTWHO_DEBUG=1, this option should be True.

`robottelo.virtwho_utils.restart_virtwho_service` ()

Do the following: 1. remove rhsm.log to ensure there are no old messages. 2. restart virt-who service via systemctl command

`robottelo.virtwho_utils.update_configure_option` (*option, value, config_file*)

Update option in virt-who config file :param option: the option you want to update :param value: set the option to the value :param config_file: path of virt-who config file

`robottelo.virtwho_utils.delete_configure_option` (*option, config_file*)

Delete option in virt-who config file :param option: the option you want to delete :param config_file: path of virt-who config file

`robottelo.virtwho_utils.add_configure_option` (*option, value, config_file*)

Add option to virt-who config file :param option: the option you want to add :param value: the value of the option :param config_file: path of virt-who config file

`robottelo.virtwho_utils.hypervisor_json_create` (*hypervisors, guests*)

Create a hypervisor guest json data. For example: {'hypervisors': [{'hypervisorId': '820b5143-3885-4dba-9358-4ce8c30d934e', 'guests': [{'guestId': 'afb91b1f-8438-46f5-bc67-d7ab328ef782', 'state': 1, 'attributes': {'active': 1, 'virtWhoType': 'esx'}}]}]} :param hypervisors: how many hypervisors will be created :param guests: how many guests will be created

`robottelo.virtwho_utils.create_fake_hypervisor_content` (*org_label, hypervisors, guests*)

Post the fake hypervisor content to satellite server :param hypervisors: how many hypervisors will be created :param guests: how many guests will be created :param org_label: the label of the Organization :return data: the hypervisor content

`robottelo.virtwho_utils.get_hypervisor_info` ()

Get the hypervisor_name and guest_name from rhsm.log.

`robottelo.virtwho_utils.virtwho_package_locked` ()

Uninstall virt-who package and lock the foreman-maintain packages.

robottelo.vm

Utilities to create clients

Clients are virtual machines provisioned on a provisioning_server. All virtual machine images are stored on the image_dir path on the provisioning server.

Make sure to configure the clients section on the configuration file. Also make sure that the server have in place: the base images for rhel66 and rhel71, snap-guest and its dependencies and the image_dir path created.

Module Contents

Classes

<i>VirtualMachine</i> (cpu=1, ram=512, distro=None, provisioning_server=None, image_dir=None, tag=None, hostname=None, domain=None, source_image=None, target_image=None, bridge=None, network=None)	Manages a virtual machine to allow client provisioning for robottelo
--	--

robottelo.vm.logger

exception robottelo.vm.VirtualMachineError

Bases: Exception

Exception raised for failed virtual machine management operations

class robottelo.vm.VirtualMachine (*cpu=1, ram=512, distro=None, provisioning_server=None, image_dir=None, tag=None, hostname=None, domain=None, source_image=None, target_image=None, bridge=None, network=None*)

Bases: object

Manages a virtual machine to allow client provisioning for robottelo

It expects that base images are created and snap-guest is setup on the provisioning server.

This also can be used as a context manager:

```
with VirtualMachine() as vm:
    result = vm.run('ls')
    out = result.stdout
```

Make sure to call *destroy()* to stop and clean the image on the provisioning server, otherwise the virtual machine and its image will stay on the server consuming hardware resources.

It is possible to customize the *provisioning_server* and *image_dir* as per virtual machine basis. Just set the wanted values when instantiating.

subscribed

domain

hostname

target_image

allowed_distros (*self*)

This is needed in construction, record it for easy reference Property instead of a class attribute to delay reading of the settings

create (*self*)

Creates a virtual machine on the provisioning server using snap-guest

Raises *robottelo.vm.VirtualMachineError* – Whenever a virtual machine could not be executed.

destroy (*self*)

Destroys the virtual machine on the provisioning server

download_install_rpm (*self, repo_url, package_name*)

Downloads and installs custom rpm on the virtual machine.

Parameters

- **repo_url** – URL to repository, where package is located.
- **package_name** – Desired package name.

Returns None.

Raises `robottelo.vm.VirtualMachineError` – If package wasn't installed.

enable_repo (*self*, *repo*, *force=False*)

Enables specified Red Hat repository on the virtual machine. Does nothing if capsule or satellite tools repo was passed and downstream with custom repo URLs detected (custom repos are enabled by default when registering a host).

Parameters

- **repo** – Red Hat repository name.
- **force** – enforce enabling command, even when custom repos are detected for satellite tools or capsule.

Returns None.

subscription_manager_list_repos (*self*)

subscription_manager_status (*self*)

create_custom_repos (*self*, ***kwargs*)

Create custom repositfiles. Each *kwargs* item will result in one repository file created. Where the key is the repository filename and repository name, and the value is the repository URL.

For example:

```
create_custom_repo(custom_repo='http://repourl.domain.com/path')
```

Will create a repository file named `custom_repo.repo` with the following contents:

```
[custom_repo]
name=custom_repo
baseurl=http://repourl.domain.com/path
enabled=1
gpgcheck=0
```

install_katello_agent (*self*)

Installs katello agent on the virtual machine.

Returns None.

Raises `robottelo.vm.VirtualMachineError` – If katello-ca wasn't installed.

install_katello_host_tools (*self*)

Installs Katello host tools on the virtual machine

Raises `robottelo.vm.VirtualMachineError` – If katello-host-tools wasn't installed.

install_katello_ca (*self*)

Downloads and installs katello-ca rpm on the virtual machine.

Uses common helper `install_katello_ca(hostname=None)`, but passes `self.ip_addr` instead of the hostname as we are using fake hostnames for virtual machines.

Returns None.

Raises `robottelo.vm.VirtualMachineError` – If katello-ca wasn't installed.

install_capsule_katello_ca (*self*, *capsule=None*)

Downloads and installs katello-ca rpm on the virtual machine.

Param str capsule: Capsule hostname

Raises *robottelo.vm.VirtualMachineError* – If katello-ca wasn't installed.

register_contenthost (*self*, *org*, *activation_key=None*, *lce=None*, *consumerid=None*, *force=True*, *releasever=None*, *username=None*, *password=None*, *auto_attach=False*)

Registers content host on foreman server using activation-key. This can be done in two ways: either by specifying organization name and activation key name or by specifying organization name and lifecycle environment name (administrator credentials for authentication will be passed automatically)

Parameters

- **activation_key** – Activation key name to register content host with.
- **lce** – lifecycle environment name to which register the content host.
- **consumerid** – uuid of content host, register to this content host, content host has to be created before
- **org** – Organization name to register content host for.
- **force** – Register the content host even if it's already registered
- **releasever** – Set a release version
- **username** – a user name to register the content host with
- **password** – the user password
- **auto_attach** – automatically attach compatible subscriptions to this system.

Returns SSHCommandResult instance filled with the result of the registration.

remove_katello_ca (*self*)

Removes katello-ca rpm from the virtual machine.

Uses common helper *remove_katello_ca(hostname=None)*, but passes *self.ip_addr* instead of the hostname as we are using fake hostnames for virtual machines.

Returns None.

Raises *robottelo.vm.VirtualMachineError* – If katello-ca wasn't removed.

remove_capsule_katello_ca (*self*, *capsule=None*)

Removes katello-ca rpm and reset rhsm.conf from the virtual machine.

Param str capsule: Capsule hostname

Raises *robottelo.vm.VirtualMachineError* – If katello-ca wasn't removed.

unregister (*self*)

Run subscription-manager unregister.

Returns SSHCommandResult instance filled with the result of the unregistration.

run (*self*, *cmd*, *timeout=None*)

Runs a ssh command on the virtual machine

Parameters

- **cmd** (*str*) – Command to run on the virtual machine
- **timeout** (*int*) – Time to wait for the ssh command to finish

Returns A `robottelo.ssh.SSHCommandResult` instance with the commands results

Return type `robottelo.ssh.SSHCommandResult`

Raises `robottelo.vm.VirtualMachineError` – If the virtual machine is not created.

get (*self*, *remote_path*, *local_path=None*)

Get a remote file from the virtual machine.

put (*self*, *local_path*, *remote_path=None*)

Put a local file to the virtual machine.

configure_rhel_repo (*self*, *rhel_repo*)

Configures specified Red Hat repository on the virtual machine.

Parameters `rhel_repo` – Red Hat repository link from properties file.

Returns None.

configure_puppet (*self*, *rhel_repo=None*, *proxy_hostname=None*)

Configures puppet on the virtual machine/Host. :param proxy_hostname: external capsule hostname
:param rhel_repo: Red Hat repository link from properties file. :return: None.

execute_foreman_scap_client (*self*, *policy_id=None*)

Executes foreman_scap_client on the vm/clients to create security audit report.

Parameters `policy_id` – The Id of the OSCAP policy.

Returns None.

configure_rhai_client (*self*, *activation_key*, *org*, *rhel_distro*)

Configures a Red Hat Access Insights service on the system by installing the redhat-access-insights package and registering to the service.

Parameters

- `activation_key` – Activation key to be used to register the system to satellite
- `org` – The org to which the system is required to be registered
- `rhel_distro` – rhel distribution for

Returns None

set_infrastructure_type (*self*, *infrastructure_type='physical'*)

Force host to appear as bare-metal or virtual machine in subscription-manager fact.

Parameters `infrastructure_type` (*str*) – One of “physical”, “virtual”

patch_os_release_version (*self*, *distro=DISTRO_RHEL7*)

Patch VM OS release version.

This is needed by yum package manager to generate the right RH repositories urls.

`__enter__` (*self*)

`__exit__` (*self*, **exc*)

`robottelo.vm_capsule`

Virtual machine client provisioning with satellite capsule product setup

Module Contents

Classes

<code>CapsuleVirtualMachine</code> (<code>cpu=4</code> , <code>ram=16384</code> , <code>distro=None</code> , <code>provisioning_server=None</code> , <code>image_dir=None</code> , <code>org_id=None</code> , <code>lce_id=None</code> , <code>organization_ids=None</code> , <code>location_ids=None</code>)	Virtual machine client provisioning with satellite capsule product
---	--

`robottelo.vm_capsule.logger`

exception `robottelo.vm_capsule.CapsuleVirtualMachineError`

Bases: `Exception`

Exception raised for failed capsule virtual machine operations

class `robottelo.vm_capsule.CapsuleVirtualMachine` (`cpu=4`, `ram=16384`, `distro=None`, `provisioning_server=None`, `image_dir=None`, `org_id=None`, `lce_id=None`, `organization_ids=None`, `location_ids=None`)

Bases: `robottelo.vm.VirtualMachine`

Virtual machine client provisioning with satellite capsule product setup

hostname_local

The virtual machine local hostname from provisioning server

capsule_org

capsule

capsule_lce

capsule_location_ids

capsule_organization_ids

`_capsule_setup_name_resolution` (*self*)

Setup a name resolution so the capsule and satellite are resolvable

`_capsule_cleanup` (*self*)

make the necessary cleanup in case of a crash

`_setup_capsule` (*self*)

Prepare the virtual machine to host a capsule node

`create` (*self*)

Creates a virtual machine on the provisioning server using snap-guest

Raises `robottelo.vm.VirtualMachineError` – Whenever a virtual machine could not be executed.

`suspend` (*self*, `ensure=False`, `timeout=None`, `connection_timeout=30`)

Suspend the virtual machine.

Parameters

- **`ensure`** (*bool*) – ensure that the virtual machine is unreachable
- **`timeout`** (*int*) – Time to wait for the ssh command to finish.

- **connection_timeout** (*int*) – Time to wait for establishing the connection.

Notes:

1. The virtual machine will consume system RAM but not processor resources. Disk and network I/O does not occur while the guest is suspended.
2. This operation is immediate and the guest can be restarted with resume.

resume (*self*, *ensure=False*, *timeout=None*, *connection_timeout=30*)

Restore from a suspended state

Parameters

- **ensure** (*bool*) – ensure that the virtual machine is reachable
- **timeout** (*int*) – Time to wait for the ssh command to finish.
- **connection_timeout** (*int*) – Time to wait for establishing the connection.

Note: This operation is immediate

destroy (*self*)

Destroys the virtual machine on the provisioning server

Want to contribute? Before submitting code, read through the [committing guide](#) and **Robottelo code standards**. Ready to start reviewing pull requests? We have [a guide](#) for that too! Finally, the [API reference](#) covers individual functions, classes, methods and modules.

Robottelo is compatible with Python 3.6+.

Bugs are listed [on GitHub](#). If you think you've found a new issue, please do one of the following:

- Open a new bug report on Github.
- Join the [#robottelo](#) IRC channel on Freenode ([irc.freenode.net](#)).

You can generate the documentation for Robottelo as follows, so long as you have [Sphinx](#) and `make` installed:

```
$ cd docs
$ make html
```

You can generate a graph of Foreman entities and their dependencies, so long as you have [graphviz](#) installed:

```
$ make graph-entities
```

To check for code smells:

```
$ pre-commit install-hooks
$ pre-commit run --all-files
```

The design and development for this software is led by [Og Maciel](#).

r

- robottelo, 22
- robottelo.api, 22
- robottelo.api.assertions, 22
- robottelo.api.utils, 23
- robottelo.cleanup, 139
- robottelo.cli, 30
- robottelo.cli.activationkey, 30
- robottelo.cli.ansible, 31
- robottelo.cli.architecture, 32
- robottelo.cli.arfreport, 32
- robottelo.cli.assertions, 33
- robottelo.cli.auth, 34
- robottelo.cli.base, 35
- robottelo.cli.capsule, 38
- robottelo.cli.computeprofile, 39
- robottelo.cli.computeresource, 40
- robottelo.cli.contentview, 41
- robottelo.cli.defaults, 43
- robottelo.cli.discoveredhost, 44
- robottelo.cli.discoveryrule, 45
- robottelo.cli.docker, 45
- robottelo.cli.domain, 49
- robottelo.cli.environment, 50
- robottelo.cli.erratum, 50
- robottelo.cli.fact, 51
- robottelo.cli.factory, 51
- robottelo.cli.file, 63
- robottelo.cli.filter, 63
- robottelo.cli.globalparam, 64
- robottelo.cli.gpgkey, 65
- robottelo.cli.hammer, 65
- robottelo.cli.host, 67
- robottelo.cli.hostcollection, 73
- robottelo.cli.hostgroup, 74
- robottelo.cli.http_proxy, 75
- robottelo.cli.job_invocation, 76
- robottelo.cli.job_template, 76
- robottelo.cli.ldapauthsource, 77
- robottelo.cli.lifecycleenvironment, 78
- robottelo.cli.location, 78
- robottelo.cli.medium, 80
- robottelo.cli.model, 81
- robottelo.cli.module_stream, 82
- robottelo.cli.operatingsys, 82
- robottelo.cli.org, 83
- robottelo.cli.ostreebranch, 85
- robottelo.cli.package, 86
- robottelo.cli.partitionable, 86
- robottelo.cli.product, 87
- robottelo.cli.proxy, 88
- robottelo.cli.puppet, 89
- robottelo.cli.puppetmodule, 90
- robottelo.cli.realm, 90
- robottelo.cli.recurring_logic, 91
- robottelo.cli.report, 91
- robottelo.cli.report_template, 92
- robottelo.cli.repository, 93
- robottelo.cli.repository_set, 94
- robottelo.cli.role, 95
- robottelo.cli.scap_policy, 96
- robottelo.cli.scap_tailoring_files, 96
- robottelo.cli.scapcontent, 97
- robottelo.cli.scpams, 98
- robottelo.cli.settings, 99
- robottelo.cli.srpm, 99
- robottelo.cli.subnet, 100
- robottelo.cli.subscription, 101
- robottelo.cli.syncplan, 101
- robottelo.cli.task, 102
- robottelo.cli.template, 103
- robottelo.cli.template_input, 104
- robottelo.cli.template_sync, 104
- robottelo.cli.user, 105
- robottelo.cli.usergroup, 106
- robottelo.cli.virt_who_config, 108
- robottelo.config, 109
- robottelo.config.base, 109
- robottelo.config.cast, 118

- robottelo.constants, 140
- robottelo.datafactory, 147
- robottelo.decorators, 120
- robottelo.decorators.func_locker, 127
- robottelo.decorators.func_shared, 120
- robottelo.decorators.func_shared.base, 120
- robottelo.decorators.func_shared.file_storage, 121
- robottelo.decorators.func_shared.redis_storage, 122
- robottelo.decorators.func_shared.shared, 123
- robottelo.decorators.host, 130
- robottelo.errors, 151
- robottelo.helpers, 151
- robottelo.host_info, 157
- robottelo.libvirt_discovery, 158
- robottelo.log, 159
- robottelo.manifests, 159
- robottelo.products, 162
- robottelo.report_portal, 133
- robottelo.report_portal.portal, 133
- robottelo.rhssso_utils, 171
- robottelo.ssh, 172
- robottelo.system_facts, 177
- robottelo.test, 177
- robottelo.ui, 135
- robottelo.ui.utils, 135
- robottelo.upgrade_utility, 180
- robottelo.utils, 135
- robottelo.utils.issue_handlers, 136
- robottelo.utils.issue_handlers.bugzilla, 136
- robottelo.utils.version, 138
- robottelo.virtwho_utils, 181
- robottelo.vm, 184
- robottelo.vm_capsule, 188

Symbols

- `_AssertNotRaisesContext` (class in `robot-telo.test`), 178
- `_DATETIME_FORMAT` (in module `robot-telo.decorators.func_shared.shared`), 125
- `_DEFAULT_CLASS_NAME_DEPTH` (in module `robot-telo.decorators.func_locker`), 129
- `_DEFAULT_CLASS_NAME_DEPTH` (in module `robot-telo.decorators.func_shared.shared`), 125
- `_NAMESPACE_SCOPE_KEY_TYPE` (in module `robot-telo.decorators.func_shared.shared`), 125
- `_SAT_6_1_VERSION_COMMAND` (in module `robot-telo.host_info`), 157
- `_SAT_6_2_VERSION_COMMAND` (in module `robot-telo.host_info`), 157
- `_SERVER_CERT_MD5` (in module `robot-telo.decorators.func_shared.shared`), 125
- `_STATE_FAILED` (in module `robot-telo.decorators.func_shared.shared`), 125
- `_STATE_READY` (in module `robot-telo.decorators.func_shared.shared`), 125
- `_SharedFunction` (class in `robot-telo.decorators.func_shared.shared`), 126
- `__call__()` (`robottelo.config.casts.Boolean` method), 118
- `__call__()` (`robottelo.config.casts.Dict` method), 119
- `__call__()` (`robottelo.config.casts.List` method), 119
- `__call__()` (`robottelo.config.casts.LoggingLevel` method), 119
- `__call__()` (`robottelo.config.casts.Tuple` method), 119
- `__call__()` (`robottelo.config.casts.WebdriverDesiredCapabilities` method), 119
- `__call__()` (`robottelo.decorators.func_shared.shared.SharedFunction` method), 126
- `__call__()` (`robottelo.helpers.ServerFileDownloader` method), 153
- `__enter__()` (`robottelo.api.utils.templateupdate` method), 28
- `__enter__()` (`robottelo.libvirt_discovery.LibvirtGuest` method), 159
- `__enter__()` (`robottelo.manifests.Manifest` method), 161
- `__enter__()` (`robottelo.test._AssertNotRaisesContext` method), 178
- `__enter__()` (`robottelo.vm.VirtualMachine` method), 188
- `__exit__()` (`robottelo.api.utils.templateupdate` method), 29
- `__exit__()` (`robottelo.libvirt_discovery.LibvirtGuest` method), 159
- `__exit__()` (`robottelo.manifests.Manifest` method), 161
- `__exit__()` (`robottelo.test._AssertNotRaisesContext` method), 178
- `__exit__()` (`robottelo.vm.VirtualMachine` method), 188
- `__getitem__()` (`robottelo.host_info.SatVersionDependentValues` method), 158
- `__iter__()` (`robottelo.products.RepositoryCollection` method), 170
- `__repr__()` (`robottelo.cli.base.CLIBaseError` method), 35
- `__repr__()` (`robottelo.products.BaseRepository` method), 167
- `__repr__()` (`robottelo.products.GenericRHRRepository` method), 168
- `__repr__()` (`robottelo.ssh.SSHCommandResult` method), 173
- `__str__()` (`robottelo.cli.base.CLIBaseError` method), 35
- `_abcfs` (in module `robottelo.constants`), 140
- `_abcs` (in module `robottelo.constants`), 140
- `_bcds` (in module `robottelo.constants`), 140
- `_bios_date()` (in module `robottelo.system_facts`), 177
- `_booleans` (`robottelo.config.casts.Boolean` attribute),

118

`_call_function()` (*robot-telo.decorators.func_shared.shared._SharedFunction method*), 126

`_call_paramiko_sshclient()` (*in module robot-telo.ssh*), 173

`_capsule_cleanup()` (*robot-telo.vm_capsule.CapsuleVirtualMachine method*), 189

`_capsule_setup_name_resolution()` (*robot-telo.vm_capsule.CapsuleVirtualMachine method*), 189

`_check_config()` (*in module robot-telo.decorators.func_shared.shared*), 125

`_check_deadlock()` (*in module robot-telo.decorators.func_locker*), 129

`_configure_airgun()` (*robottelo.config.Settings method*), 120

`_configure_airgun()` (*robot-telo.config.base.Settings method*), 118

`_configure_entities()` (*robottelo.config.Settings method*), 120

`_configure_entities()` (*robot-telo.config.base.Settings method*), 117

`_configure_logging()` (*robottelo.config.Settings method*), 120

`_configure_logging()` (*robot-telo.config.base.Settings method*), 118

`_configure_third_party_logging()` (*robot-telo.config.Settings method*), 120

`_configure_third_party_logging()` (*robot-telo.config.base.Settings method*), 118

`_configured` (*in module robot-telo.decorators.func_shared.shared*), 125

`_construct_command()` (*robottelo.cli.base.Base class method*), 38

`_csv_reader()` (*in module robottelo.cli.hammer*), 66

`_custom_product_info` (*robot-telo.products.RepositoryCollection attribute*), 169

`_db_error_regex` (*robottelo.cli.base.Base attribute*), 37

`_default_interface` (*robottelo.test.APITestCase attribute*), 180

`_default_interface` (*robottelo.test.CLITestCase attribute*), 180

`_default_interface` (*robottelo.test.TestCase attribute*), 179

`_default_notraises_value_handler` (*robot-telo.test.APITestCase attribute*), 180

`_default_notraises_value_handler` (*robot-telo.test.CLITestCase attribute*), 180

`_default_notraises_value_handler` (*robot-telo.test.TestCase attribute*), 179

`_distro` (*robottelo.products.BaseRepository attribute*), 167

`_distro` (*robottelo.products.GenericRHRepository attribute*), 168

`_distro` (*robottelo.products.RHELCloudFormsTools attribute*), 169

`_distro` (*robottelo.products.RepositoryCollection attribute*), 169

`_distro` (*robottelo.products.VirtualizationAgentsRepository attribute*), 169

`_download_manifest_info()` (*robot-telo.manifests.ManifestCloner method*), 160

`_encode_result_kwargs()` (*robot-telo.decorators.func_shared.shared._SharedFunction method*), 126

`_entity_with_credentials()` (*in module robot-telo.cli.factory*), 54

`_extract_sat_version()` (*in module robot-telo.host_info*), 157

`_format_launches()` (*robot-telo.report_portal.portal.ReportPortal method*), 134

`_gen_mac_for_libvirt()` (*in module robot-telo.libvirt_discovery*), 158

`_get_capsule_vm_distro_repos()` (*in module robottelo.cli.factory*), 61

`_get_default_scope()` (*in module robot-telo.decorators.func_locker*), 129

`_get_default_scope()` (*in module robot-telo.decorators.func_shared.shared*), 125

`_get_default_storage_handler()` (*in module robottelo.decorators.func_shared.shared*), 125

`_get_function_name()` (*in module robot-telo.decorators.func_locker*), 129

`_get_function_name()` (*in module robot-telo.decorators.func_shared.shared*), 126

`_get_function_name_key()` (*in module robot-telo.decorators.func_shared.shared*), 126

`_get_function_name_lock_path()` (*in module robottelo.decorators.func_locker*), 129

`_get_hypervisor_mapping()` (*in module robot-telo.virtwho_utils*), 183

`_get_kwargs_md5()` (*in module robot-telo.decorators.func_shared.shared*), 126

`_get_repo_data()` (*robot-telo.products.GenericRHRepository method*), 168

`_get_root_dir()` (*in module robot-telo.decorators.func_shared.file_storage*), 121

`_get_scope_name()` (*in module robot-telo.decorators.func_shared.shared*), 126

`_get_scope_path()` (*in module robot-*

telo.decorators.func_locker), 129

`_get_temp_lock_function_dir()` (in module *robottelo.decorators.func_locker*), 129

`_get_username_password()` (*robottelo.cli.base.Base class method*), 37

`_handle_response()` (*robottelo.cli.base.Base class method*), 37

`_has_result_expired()` (*robottelo.decorators.func_shared.shared._SharedFunction method*), 126

`_items` (*robottelo.products.RepositoryCollection attribute*), 169

`_key` (*robottelo.products.GenericRHRRepository attribute*), 168

`_key` (*robottelo.products.RHELAnsibleEngineRepository attribute*), 169

`_key` (*robottelo.products.RHELCloudFormsTools attribute*), 169

`_key` (*robottelo.products.RHELRepository attribute*), 169

`_key` (*robottelo.products.SatelliteCapsuleRepository attribute*), 169

`_key` (*robottelo.products.SatelliteToolsRepository attribute*), 169

`_key` (*robottelo.products.VirtualizationAgentsRepository attribute*), 169

`_launch_requester()` (*robottelo.report_portal.portal.ReportPortal method*), 134

`_logging_levels` (*robottelo.config.casts.LoggingLevel attribute*), 119

`_manifest_cloner` (in module *robottelo.manifests*), 160

`_multiprocess_can_split_` (*robottelo.test.APITestCase attribute*), 180

`_multiprocess_can_split_` (*robottelo.test.CLITestCase attribute*), 180

`_normalize()` (in module *robottelo.cli.hammer*), 66

`_normalize_obj()` (in module *robottelo.cli.hammer*), 66

`_org` (*robottelo.products.RepositoryCollection attribute*), 169

`_os_repo` (*robottelo.products.RepositoryCollection attribute*), 170

`_parse_entry()` (in module *robottelo.virtwho_utils*), 182

`_read_robottelo_settings()` (*robottelo.config.Settings method*), 120

`_read_robottelo_settings()` (*robottelo.config.base.Settings method*), 117

`_repo_data` (*robottelo.products.GenericRHRRepository attribute*), 168

`_repo_info` (*robottelo.products.BaseRepository attribute*), 167

`_repo_is_distro()` (*robottelo.products.GenericRHRRepository method*), 168

`_repos_info` (*robottelo.products.RepositoryCollection attribute*), 169

`_server_distro` (in module *robottelo.products*), 166

`_set_configured()` (in module *robottelo.decorators.func_shared.shared*), 125

`_setup_capsule()` (*robottelo.vm_capsule.CapsuleVirtualMachine method*), 189

`_setup_content_data` (*robottelo.products.RepositoryCollection attribute*), 170

`_setup_org_for_a_rh_repo()` (in module *robottelo.cli.factory*), 59

`_storage_handlers` (in module *robottelo.decorators.func_shared.shared*), 125

`_test_params()` (*robottelo.report_portal.portal.Launch method*), 135

`_test_requester()` (*robottelo.report_portal.portal.Launch method*), 135

`_type` (*robottelo.products.BaseRepository attribute*), 167

`_type` (*robottelo.products.DockerRepository attribute*), 167

`_type` (*robottelo.products.GenericRHRRepository attribute*), 168

`_type` (*robottelo.products.OSTreeRepository attribute*), 168

`_type` (*robottelo.products.PuppetRepository attribute*), 167

`_type` (*robottelo.products.YumRepository attribute*), 167

`_upload_file()` (in module *robottelo.ssh*), 175

`_url` (*robottelo.products.BaseRepository attribute*), 166

`_url` (*robottelo.products.GenericRHRRepository attribute*), 168

`_validate_registry_configs()` (*robottelo.config.base.ContainerRepositorySettings method*), 113

`_validate_robottelo_settings()` (*robottelo.config.Settings method*), 120

`_validate_robottelo_settings()` (*robottelo.config.base.Settings method*), 117

`_versions()` (*robottelo.report_portal.portal.Launch method*), 134

`_write_content()` (in module *robottelo.decorators.func_locker*), 129

`_zones_combo` (in module *robottelo.constants*), 140

A

- ActivationKey (class in *robottelo.cli.activationkey*), 30
- activationkey_add_subscription_to_repo() (in module *robottelo.cli.factory*), 59
- add() (*robottelo.cli.defaults.Defaults* class method), 44
- add_architecture() (*robottelo.cli.operatingsys.OperatingSys* class method), 83
- add_authorized_key() (in module *robottelo.ssh*), 175
- add_compute_resource() (*robottelo.cli.location.Location* class method), 79
- add_compute_resource() (*robottelo.cli.org.Org* class method), 84
- add_configure_option() (in module *robottelo.virtwho_utils*), 184
- add_domain() (*robottelo.cli.location.Location* class method), 79
- add_domain() (*robottelo.cli.org.Org* class method), 84
- add_environment() (*robottelo.cli.location.Location* class method), 79
- add_environment() (*robottelo.cli.org.Org* class method), 84
- add_host() (*robottelo.cli.hostcollection.HostCollection* class method), 74
- add_host_collection() (*robottelo.cli.activationkey.ActivationKey* class method), 31
- add_hostgroup() (*robottelo.cli.location.Location* class method), 79
- add_hostgroup() (*robottelo.cli.org.Org* class method), 84
- add_item() (*robottelo.products.RepositoryCollection* method), 170
- add_items() (*robottelo.products.RepositoryCollection* method), 170
- add_location() (*robottelo.cli.org.Org* class method), 85
- add_matcher() (*robottelo.cli.sparams.SmartClassParameter* class method), 98
- add_medium() (*robottelo.cli.location.Location* class method), 79
- add_medium() (*robottelo.cli.org.Org* class method), 85
- add_operating_system() (*robottelo.cli.base.Base* class method), 37
- add_operatingsystem() (*robottelo.cli.template.Template* class method), 104
- add_organization() (*robottelo.cli.location.Location* class method), 79
- add_provisioning_template() (*robottelo.cli.location.Location* class method), 79
- add_provisioning_template() (*robottelo.cli.operatingsys.OperatingSys* class method), 83
- add_provisioning_template() (*robottelo.cli.org.Org* class method), 85
- add_ptable() (*robottelo.cli.operatingsys.OperatingSys* class method), 83
- add_remote_execution_ssh_key() (in module *robottelo.helpers*), 154
- add_repository() (*robottelo.cli.contentview.ContentView* class method), 42
- add_role() (*robottelo.cli.user.User* class method), 106
- add_role() (*robottelo.cli.usergroup.UserGroup* class method), 107
- add_role_permissions() (in module *robottelo.cli.factory*), 61
- add_smart_proxy() (*robottelo.cli.location.Location* class method), 80
- add_smart_proxy() (*robottelo.cli.org.Org* class method), 85
- add_subnet() (*robottelo.cli.location.Location* class method), 80
- add_subnet() (*robottelo.cli.org.Org* class method), 85
- add_subscription() (*robottelo.cli.activationkey.ActivationKey* class method), 31
- add_to_content_view() (*robottelo.products.BaseRepository* method), 167
- add_to_content_view() (*robottelo.products.PuppetRepository* method), 167
- add_uppercase_char_into_string() (in module *robottelo.datafactory*), 148
- add_user() (*robottelo.cli.location.Location* class method), 80
- add_user() (*robottelo.cli.org.Org* class method), 85
- add_user() (*robottelo.cli.usergroup.UserGroup* class method), 107
- add_user_group() (*robottelo.cli.usergroup.UserGroup* class method), 107
- add_version() (*robottelo.cli.contentview.ContentView* class method), 42

- `add_workaround()` (in module `robottelo.utils.issue_handlers`), 137
- `all_features` (`robottelo.config.base.Settings` attribute), 117
- `all_features` (`robottelo.config.Settings` attribute), 119
- `allowed_distros()` (`robottelo.vm.VirtualMachine` method), 185
- Ansible (class in `robottelo.cli.ansible`), 31
- `ansible_roles_play()` (`robottelo.cli.host.Host` class method), 68
- ANSWERS (in module `robottelo.constants`), 146
- ANY_CONTEXT (in module `robottelo.constants`), 146
- `api_url` (`robottelo.report_portal.portal.ReportPortal` attribute), 134
- `APINotRaisesValueHandler` (class in `robottelo.test`), 178
- `APITestCase` (class in `robottelo.test`), 180
- `apply_package_filter()` (in module `robottelo.api.utils`), 29
- Architecture (class in `robottelo.cli.architecture`), 32
- ARCHITECTURES (in module `robottelo.system_facts`), 177
- `Arfreport` (class in `robottelo.cli.arfreport`), 33
- `assert_api_not_raises()` (in module `robottelo.api.assertions`), 22
- `assert_api_not_raises_regex()` (in module `robottelo.api.assertions`), 23
- `assert_cli_not_raises()` (in module `robottelo.cli.assertions`), 34
- `assert_cli_not_raises_regex()` (in module `robottelo.cli.assertions`), 34
- `assert_error_msg()` (`robottelo.test.CLITestCase` method), 180
- `AssertApiNotRaisesContextManager` (class in `robottelo.test`), 179
- `AssertCliNotRaisesContextManager` (class in `robottelo.test`), 178
- `assertNotRaises()` (`robottelo.test.TestCase` method), 179
- `assertNotRaisesRegex()` (`robottelo.test.TestCase` method), 180
- ATOMIC_HOST_TREE (in module `robottelo.constants`), 142
- `attach_custom_product_subscription()` (in module `robottelo.api.utils`), 28
- `attach_nic()` (`robottelo.libvirt_discovery.LibvirtGuest` method), 159
- AUDIENCE_MAPPER (in module `robottelo.constants`), 146
- Auth (class in `robottelo.cli.auth`), 34
- AuthLogin (class in `robottelo.cli.auth`), 35
- `available_permissions()` (`robottelo.cli.filter.Filter` class method), 64
- `available_repositories()` (`robottelo.cli.repository_set.RepositorySet` class method), 94
- AWS_EC2_FLAVOR_T2_MICRO (in module `robottelo.constants`), 140
- AZURERM_FILE_URI (in module `robottelo.constants`), 141
- AZURERM_PLATFORM_DEFAULT (in module `robottelo.constants`), 141
- AZURERM_PREMIUM_OS_Disk (in module `robottelo.constants`), 141
- AZURERM_RG_DEFAULT (in module `robottelo.constants`), 141
- AZURERM_RHEL7_FT_BYOS_IMG_URN (in module `robottelo.constants`), 141
- AZURERM_RHEL7_FT_CUSTOM_IMG_URN (in module `robottelo.constants`), 141
- AZURERM_RHEL7_FT_GALLERY_IMG_URN (in module `robottelo.constants`), 141
- AZURERM_RHEL7_FT_IMG_URN (in module `robottelo.constants`), 141
- AZURERM_RHEL7_UD_IMG_URN (in module `robottelo.constants`), 141
- AZURERM_VALID_REGIONS (in module `robottelo.constants`), 141
- AZURERM_VM_SIZE_DEFAULT (in module `robottelo.constants`), 141
- `AzureRMSettings` (class in `robottelo.config.base`), 113
- ## B
- BACKUP_FILES (in module `robottelo.constants`), 146
- Base (class in `robottelo.cli.base`), 36
- BaseRepository (class in `robottelo.products`), 166
- BaseStorageHandler (class in `robottelo.decorators.func_shared.base`), 121
- `basic()` (`robottelo.cli.auth.AuthLogin` class method), 35
- BOOKMARK_ENTITIES (in module `robottelo.constants`), 146
- Boolean (class in `robottelo.config.casts`), 118
- `BugzillaSettings` (class in `robottelo.config.base`), 112
- `build_pxe_default()` (`robottelo.cli.template.Template` class method), 104
- ## C
- `cacheable()` (in module `robottelo.decorators`), 133
- CACHED_RESPONSES (in module `robottelo.utils.issue_handlers.bugzilla`), 137
- `call_entity_method_with_timeout()` (in module `robottelo.api.utils`), 24

- Capsule (class in *robottelo.cli.capsule*), 39
- capsule (*robottelo.vm_capsule.CapsuleVirtualMachine* attribute), 189
- capsule_cleanup() (in module *robottelo.cleanup*), 139
- capsule_lce (*robottelo.vm_capsule.CapsuleVirtualMachine* attribute), 189
- capsule_location_ids (*robottelo.vm_capsule.CapsuleVirtualMachine* attribute), 189
- capsule_org (*robottelo.vm_capsule.CapsuleVirtualMachine* attribute), 189
- capsule_organization_ids (*robottelo.vm_capsule.CapsuleVirtualMachine* attribute), 189
- CapsuleSettings (class in *robottelo.config.base*), 112
- CapsuleTunnelError, 89
- CapsuleVirtualMachine (class in *robottelo.vm_capsule*), 189
- CapsuleVirtualMachineError, 189
- cast_boolean (*robottelo.config.base.INIReader* attribute), 110
- cast_dict (*robottelo.config.base.INIReader* attribute), 110
- cast_list (*robottelo.config.base.INIReader* attribute), 110
- cast_logging_level (*robottelo.config.base.INIReader* attribute), 110
- cast_tuple (*robottelo.config.base.INIReader* attribute), 110
- cast_webdriver_desired_capabilities (*robottelo.config.base.INIReader* attribute), 110
- cdn (*robottelo.products.BaseRepository* attribute), 167
- cdn (*robottelo.products.GenericRHRepository* attribute), 168
- CERT_PATH (in module *robottelo.constants*), 143
- CertsSettings (class in *robottelo.config.base*), 112
- check_create_os_with_title() (in module *robottelo.api.utils*), 28
- check_package_installed() (in module *robottelo.upgrade_utility*), 181
- CHECKSUM_TYPE (in module *robottelo.constants*), 141
- cleanup_of_provisioned_server() (in module *robottelo.cleanup*), 139
- CLIBaseError, 35
- CLIDataBaseError, 36
- client (*robottelo.decorators.func_shared.redis_storage.RedisStorage* attribute), 122
- ClientsSettings (class in *robottelo.config.base*), 112
- CLIFactoryError, 53
- CLINotRaisesValueHandler (class in *robottelo.test*), 178
- CLIReturnCodeError, 35
- CLITestCase (class in *robottelo.test*), 180
- clone() (in module *robottelo.manifests*), 161
- clone() (*robottelo.cli.report_template.ReportTemplate* class method), 92
- clone() (*robottelo.cli.role.Role* class method), 95
- clone() (*robottelo.cli.template.Template* class method), 104
- clone() (*robottelo.manifests.ManifestCloner* method), 160
- CLOSED_STATUSES (in module *robottelo.constants*), 146
- collect_clones() (in module *robottelo.utils.issue_handlers.bugzilla*), 137
- collect_data_bz() (in module *robottelo.utils.issue_handlers.bugzilla*), 137
- collect_dupes() (in module *robottelo.utils.issue_handlers.bugzilla*), 137
- command() (in module *robottelo.ssh*), 176
- command_base (*robottelo.cli.activationkey.ActivationKey* attribute), 30
- command_base (*robottelo.cli.ansible.Ansible* attribute), 31
- command_base (*robottelo.cli.architecture.Architecture* attribute), 32
- command_base (*robottelo.cli.arfreport.Arfreport* attribute), 33
- command_base (*robottelo.cli.auth.Auth* attribute), 34
- command_base (*robottelo.cli.auth.AuthLogin* attribute), 35
- command_base (*robottelo.cli.base.Base* attribute), 37
- command_base (*robottelo.cli.capsule.Capsule* attribute), 39
- command_base (*robottelo.cli.computeprofile.ComputeProfile* attribute), 40
- command_base (*robottelo.cli.computeresource.ComputeResource* attribute), 40
- command_base (*robottelo.cli.contentview.ContentView* attribute), 42
- command_base (*robottelo.cli.contentview.ContentViewFilter* attribute), 42
- command_base (*robottelo.cli.contentview.ContentViewFilterRule* attribute), 42
- command_base (*robottelo.cli.defaults.Defaults* attribute), 44

- command_base (robot-telo.cli.discoveredhost.DiscoveredHost attribute), 45
- command_base (robot-telo.cli.discoveryrule.DiscoveryRule attribute), 45
- command_base (robottelo.cli.docker.Docker attribute), 49
- command_base (robottelo.cli.docker.DockerManifest attribute), 46
- command_base (robottelo.cli.docker.DockerTag attribute), 47
- command_base (robottelo.cli.domain.Domain attribute), 49
- command_base (robot-telo.cli.environment.Environment attribute), 50
- command_base (robottelo.cli.erratum.Erratum attribute), 51
- command_base (robottelo.cli.fact.Fact attribute), 51
- command_base (robottelo.cli.file.File attribute), 63
- command_base (robottelo.cli.filter.Filter attribute), 64
- command_base (robot-telo.cli.globalparam.GlobalParameter attribute), 65
- command_base (robottelo.cli.gpgkey.GPGKey attribute), 65
- command_base (robottelo.cli.host.Host attribute), 68
- command_base (robottelo.cli.host.HostInterface attribute), 73
- command_base (robot-telo.cli.hostcollection.HostCollection attribute), 74
- command_base (robottelo.cli.hostgroup.HostGroup attribute), 75
- command_base (robottelo.cli.http_proxy.HttpProxy attribute), 76
- command_base (robot-telo.cli.job_invocation.JobInvocation attribute), 76
- command_base (robot-telo.cli.job_template.JobTemplate attribute), 77
- command_base (robot-telo.cli.ldapauthsource.LDAPAuthSource attribute), 77
- command_base (robot-telo.cli.lifecycleenvironment.LifecycleEnvironment attribute), 78
- command_base (robottelo.cli.location.Location attribute), 79
- command_base (robottelo.cli.medium.Medium attribute), 81
- command_base (robottelo.cli.model.Model attribute), 81
- command_base (robot-telo.cli.module_stream.ModuleStream attribute), 82
- command_base (robot-telo.cli.operatingsys.OperatingSys attribute), 83
- command_base (robottelo.cli.org.Org attribute), 84
- command_base (robot-telo.cli.ostreebranch.OstreeBranch attribute), 86
- command_base (robottelo.cli.package.Package attribute), 86
- command_base (robot-telo.cli.partitiontable.PartitionTable attribute), 87
- command_base (robottelo.cli.product.Product attribute), 88
- command_base (robottelo.cli.proxy.Proxy attribute), 89
- command_base (robottelo.cli.puppet.Puppet attribute), 89
- command_base (robot-telo.cli.puppetmodule.PuppetModule attribute), 90
- command_base (robottelo.cli.realm.Realm attribute), 91
- command_base (robot-telo.cli.recurring_logic.RecurringLogic attribute), 91
- command_base (robottelo.cli.report.Report attribute), 92
- command_base (robot-telo.cli.report_template.ReportTemplate attribute), 92
- command_base (robottelo.cli.repository.Repository attribute), 93
- command_base (robot-telo.cli.repository_set.RepositorySet attribute), 94
- command_base (robottelo.cli.role.Role attribute), 95
- command_base (robottelo.cli.scap_policy.Scappolicy attribute), 96
- command_base (robot-telo.cli.scap_tailoring_files.TailoringFiles attribute), 97
- command_base (robottelo.cli.scapcontent.Scapcontent attribute), 97
- command_base (robot-telo.cli.scparams.SmartClassParameter attribute), 98
- command_base (robottelo.cli.settings.Settings attribute), 99
- command_base (robottelo.cli.srpm.Srpm attribute), 99

- 100
- command_base (*robottelo.cli.subnet.Subnet* attribute), 100
- command_base (*robottelo.cli.subscription.Subscription* attribute), 101
- command_base (*robottelo.cli.syncplan.SyncPlan* attribute), 102
- command_base (*robottelo.cli.task.Task* attribute), 103
- command_base (*robottelo.cli.template.Template* attribute), 103
- command_base (*robottelo.cli.template_input.TemplateInput* attribute), 104
- command_base (*robottelo.cli.user.User* attribute), 106
- command_base (*robottelo.cli.usergroup.UserGroup* attribute), 107
- command_base (*robottelo.cli.usergroup.UserGroupExternal* attribute), 108
- command_base (*robottelo.cli.virt_who_config.VirtWhoConfig* attribute), 108
- command_requires_org (*robottelo.cli.base.Base* attribute), 37
- command_requires_org (*robottelo.cli.gpgkey.GPGKey* attribute), 65
- command_requires_org (*robottelo.cli.lifecycleenvironment.LifecycleEnvironment* attribute), 78
- command_requires_org (*robottelo.cli.product.Product* attribute), 88
- command_requires_org (*robottelo.cli.repository.Repository* attribute), 93
- command_sub (*robottelo.cli.base.Base* attribute), 37
- command_timeout (*robottelo.config.base.SSHClientSettings* attribute), 116
- component_add() (*robottelo.cli.contentview.ContentView* class method), 43
- component_list() (*robottelo.cli.contentview.ContentView* class method), 43
- COMPUTE_PROFILE_LARGE (in module *robottelo.constants*), 140
- COMPUTE_PROFILE_SMALL (in module *robottelo.constants*), 140
- ComputeProfile (class in *robottelo.cli.computeprofile*), 40
- ComputeResource (class in *robottelo.cli.computeresource*), 40
- configure() (*robottelo.config.base.Settings* method), 117
- configure() (*robottelo.config.Settings* method), 119
- configure_env_for_provision() (in module *robottelo.cli.factory*), 60
- configure_provisioning() (in module *robottelo.api.utils*), 26
- configure_puppet() (*robottelo.vm.VirtualMachine* method), 188
- configure_rhai_client() (*robottelo.vm.VirtualMachine* method), 188
- configure_rhel_repo() (*robottelo.vm.VirtualMachine* method), 188
- configured (*robottelo.config.base.Settings* attribute), 117
- configured (*robottelo.config.Settings* attribute), 119
- connection_timeout (*robottelo.config.base.SSHClientSettings* attribute), 116
- ContainerRepositorySettings (class in *robottelo.config.base*), 112
- content (*robottelo.manifests.Manifest* attribute), 161
- content_add_lifecycle_environment() (*robottelo.cli.capsule.Capsule* class method), 39
- content_available_lifecycle_environments() (*robottelo.cli.capsule.Capsule* class method), 39
- CONTENT_CREDENTIALS_TYPES (in module *robottelo.constants*), 140
- content_info() (*robottelo.cli.capsule.Capsule* class method), 39
- content_lifecycle_environments() (*robottelo.cli.capsule.Capsule* class method), 39
- content_override() (*robottelo.cli.activationkey.ActivationKey* class method), 31
- content_remove_lifecycle_environment() (*robottelo.cli.capsule.Capsule* class method), 39
- content_synchronization_status() (*robottelo.cli.capsule.Capsule* class method), 39
- content_synchronize() (*robottelo.cli.capsule.Capsule* class method), 39
- content_type (*robottelo.products.BaseRepository* attribute), 167
- CONTENT_VIEW_KEYS (in module *robottelo.cli.factory*), 53
- ContentView (class in *robottelo.cli.contentview*), 42
- ContentViewFilter (class in *robottelo.cli.contentview*), 42
- ContentViewFilterRule (class in *robottelo.cli.contentview*), 42
- copy() (*robottelo.cli.activationkey.ActivationKey* class method), 31

- copy () (*robottelo.cli.contentview.ContentView class method*), 42
 - copy () (*robottelo.cli.hostcollection.HostCollection class method*), 74
 - create () (*robottelo.cli.base.Base class method*), 37
 - create () (*robottelo.cli.contentview.ContentViewFilterRule class method*), 42
 - create () (*robottelo.cli.host.HostInterface class method*), 73
 - create () (*robottelo.cli.report_template.ReportTemplate class method*), 92
 - create () (*robottelo.cli.repository.Repository class method*), 93
 - create () (*robottelo.cli.template_input.TemplateInput class method*), 104
 - create () (*robottelo.cli.usergroup.UserGroupExternal class method*), 108
 - create () (*robottelo.libvirt_discovery.LibvirtGuest method*), 159
 - create () (*robottelo.products.BaseRepository method*), 167
 - create () (*robottelo.products.DockerRepository method*), 167
 - create () (*robottelo.products.GenericRHRepository method*), 168
 - create () (*robottelo.vm.VirtualMachine method*), 185
 - create () (*robottelo.vm_capsule.CapsuleVirtualMachine method*), 189
 - create_custom_repos () (*robottelo.vm.VirtualMachine method*), 186
 - create_discovered_host () (*in module robottelo.api.utils*), 28
 - create_fake_host () (*in module robottelo.ui.utils*), 135
 - create_fake_hypervisor_content () (*in module robottelo.virtwho_utils*), 184
 - create_mapper () (*in module robottelo.rhssso_utils*), 172
 - create_new_rhssso_user () (*in module robottelo.rhssso_utils*), 172
 - create_object () (*in module robottelo.cli.factory*), 54
 - create_org_admin_role () (*in module robottelo.api.utils*), 29
 - create_org_admin_user () (*in module robottelo.api.utils*), 29
 - create_repo () (*in module robottelo.helpers*), 156
 - create_repo () (*in module robottelo.upgrade_utility*), 181
 - create_role_permissions () (*in module robottelo.api.utils*), 26
 - create_sync_custom_repo () (*in module robottelo.api.utils*), 25
 - CUSTOM_FILE_REPO (*in module robottelo.constants*), 142
 - CUSTOM_FILE_REPO_FILES_COUNT (*in module robottelo.constants*), 143
 - CUSTOM_KICKSTART_REPO (*in module robottelo.constants*), 143
 - CUSTOM_LOCAL_FILE (*in module robottelo.constants*), 143
 - CUSTOM_LOCAL_FOLDER (*in module robottelo.constants*), 142
 - CUSTOM_MODULE_STREAM_REPO_1 (*in module robottelo.constants*), 143
 - CUSTOM_MODULE_STREAM_REPO_2 (*in module robottelo.constants*), 143
 - custom_product (*robottelo.products.RepositoryCollection attribute*), 170
 - CUSTOM_PUPPET_REPO (*in module robottelo.constants*), 143
 - CUSTOM_REPODATA_PATH (*in module robottelo.constants*), 143
 - custom_repos (*robottelo.products.RepositoryCollection attribute*), 170
 - custom_repos_info (*robottelo.products.RepositoryCollection attribute*), 170
 - CUSTOM_RPM_REPO (*in module robottelo.constants*), 143
 - CUSTOM_RPM_SHA_512 (*in module robottelo.constants*), 143
 - CUSTOM_RPM_SHA_512_FEED_COUNT (*in module robottelo.constants*), 143
 - CUSTOM_SWID_TAG_REPO (*in module robottelo.constants*), 143
 - cv_publish_promote () (*in module robottelo.api.utils*), 26
- ## D
- data (*robottelo.products.BaseRepository attribute*), 167
 - data (*robottelo.products.GenericRHRepository attribute*), 168
 - DataFileError, 152
 - decode () (*robottelo.decorators.func_shared.base.BaseStorageHandler static method*), 121
 - decode_to_utf8 () (*in module robottelo.ssh*), 173
 - default () (*robottelo.utils.version.VersionEncoder method*), 138
 - DEFAULT_ARCHITECTURE (*in module robottelo.constants*), 142
 - DEFAULT_ATOMIC_TEMPLATE (*in module robottelo.constants*), 142
 - DEFAULT_CALL_RETRIES (*in module robottelo.decorators.func_shared.shared*), 125
 - DEFAULT_CV (*in module robottelo.constants*), 142

- DEFAULT_LOC (in module *robottelo.constants*), 142
- DEFAULT_ORG (in module *robottelo.constants*), 142
- DEFAULT_PTABLE (in module *robottelo.constants*), 142
- DEFAULT_PXE_TEMPLATE (in module *robottelo.constants*), 142
- DEFAULT_RELEASE_VERSION (in module *robottelo.constants*), 142
- DEFAULT_ROLE (in module *robottelo.constants*), 142
- DEFAULT_STORAGE_HANDLER (in module *robottelo.decorators.func_shared.shared*), 125
- DEFAULT_SUBSCRIPTION_NAME (in module *robottelo.constants*), 142
- DEFAULT_SYSPURPOSE_ATTRIBUTES (in module *robottelo.constants*), 146
- DEFAULT_TEMPLATE (in module *robottelo.constants*), 142
- default_url_on_new_port() (in module *robottelo.helpers*), 155
- Defaults (class in *robottelo.cli.defaults*), 43
- defect_types (robottelo.report_portal.portal.ReportPortal attribute), 134
- delete() (robottelo.cli.base.Base class method), 37
- delete() (robottelo.cli.defaults.Defaults class method), 44
- delete_configure_option() (in module *robottelo.virtwho_utils*), 184
- delete_manifest() (robottelo.cli.subscription.Subscription class method), 101
- delete_parameter() (robottelo.cli.base.Base class method), 37
- delete_puppet_class() (in module *robottelo.api_utils*), 25
- delete_rhsso_user() (in module *robottelo.rhssso_utils*), 172
- deploy() (robottelo.cli.virt_who_config.VirtWhoConfig class method), 108
- deploy_configure_by_command() (in module *robottelo.virtwho_utils*), 183
- deploy_configure_by_script() (in module *robottelo.virtwho_utils*), 184
- deploy_validation() (in module *robottelo.virtwho_utils*), 183
- destroy() (robottelo.libvirt_discovery.LibvirtGuest method), 159
- destroy() (robottelo.vm.VirtualMachine method), 185
- destroy() (robottelo.vm_capsule.CapsuleVirtualMachine method), 190
- destructive (in module *robottelo.decorators*), 131
- Dict (class in *robottelo.config.casts*), 119
- disable() (robottelo.cli.repository_set.RepositorySet class method), 94
- DiscoveredHost (class in *robottelo.cli.discoveredhost*), 45
- DiscoveryISOSettings (class in *robottelo.config.base*), 115
- DiscoveryRule (class in *robottelo.cli.discoveryrule*), 45
- distro (robottelo.products.BaseRepository attribute), 167
- distro (robottelo.products.GenericRHRRepository attribute), 168
- distro (robottelo.products.RepositoryCollection attribute), 170
- DISTRO_DEFAULT (in module *robottelo.constants*), 140
- DISTRO_IDS (in module *robottelo.system_facts*), 177
- distro_major_version (robottelo.products.GenericRHRRepository attribute), 168
- DISTRO_REPOS (in module *robottelo.constants*), 142
- distro_repository (robottelo.products.GenericRHRRepository attribute), 168
- DISTRO_RHEL6 (in module *robottelo.constants*), 140
- DISTRO_RHEL7 (in module *robottelo.constants*), 140
- DISTRO_RHEL8 (in module *robottelo.constants*), 140
- DISTRO_SLES11 (in module *robottelo.constants*), 140
- DISTRO_SLES12 (in module *robottelo.constants*), 140
- DistroNotSupportedError, 166
- DISTROS_MAJOR_VERSION (in module *robottelo.constants*), 140
- DISTROS_SUPPORTED (in module *robottelo.constants*), 140
- DistroSettings (class in *robottelo.config.base*), 113
- Docker (class in *robottelo.cli.docker*), 48
- DOCKER_REGISTRY_HUB (in module *robottelo.constants*), 142
- DOCKER_RH_REGISTRY_UPSTREAM_NAME (in module *robottelo.constants*), 142
- DOCKER_UPSTREAM_NAME (in module *robottelo.constants*), 142
- DockerManifest (class in *robottelo.cli.docker*), 46
- DockerRepository (class in *robottelo.products*), 167
- DockerSettings (class in *robottelo.config.base*), 113
- DockerTag (class in *robottelo.cli.docker*), 47
- Domain (class in *robottelo.cli.domain*), 49
- DOMAIN (in module *robottelo.constants*), 141
- domain (robottelo.vm.VirtualMachine attribute), 185
- download_file() (in module *robottelo.ssh*), 176
- download_gce_cert() (in module *robottelo.helpers*), 156
- download_install_rpm() (robottelo.vm.VirtualMachine method), 185
- DOWNLOAD_POLICIES (in module *robottelo.constants*), 141
- DOWNLOAD_POLICY_BACKGROUND (in module *robottelo.products*), 166

- DOWNLOAD_POLICY_IMMEDIATE (in module *robottelo.products*), 166
- DOWNLOAD_POLICY_ON_DEMAND (in module *robottelo.products*), 166
- download_server_file (in module *robottelo.helpers*), 153
- download_tailoring_file() (*robottelo.cli.scap_tailoring_files.TailoringFiles* class method), 97
- DownloadFileError, 152
- dump() (*robottelo.cli.base.Base* class method), 37
- ## E
- EC2_REGION_CA_CENTRAL_1 (in module *robottelo.constants*), 140
- EC2Settings (class in *robottelo.config.base*), 113
- enable() (*robottelo.cli.repository_set.RepositorySet* class method), 94
- enable_repo() (*robottelo.vm.VirtualMachine* method), 186
- enable_rhrepo_and_fetchid() (in module *robottelo.api.utils*), 24
- enable_shared_function() (in module *robottelo.decorators.func_shared.shared*), 125
- enable_sync_redhat_repo() (in module *robottelo.api.utils*), 26
- ENABLED (in module *robottelo.decorators.func_shared.shared*), 125
- enc_dump() (*robottelo.cli.host.Host* class method), 68
- encode() (*robottelo.decorators.func_shared.base.BaseStorageHandler* static method), 121
- Environment (class in *robottelo.cli.environment*), 50
- ENVIRONMENT (in module *robottelo.constants*), 141
- errata_apply() (*robottelo.cli.host.Host* class method), 68
- errata_info() (*robottelo.cli.host.Host* class method), 68
- errata_list() (*robottelo.cli.host.Host* class method), 69
- Erratum (class in *robottelo.cli.erratum*), 51
- erratum_install() (*robottelo.cli.hostcollection.HostCollection* class method), 74
- escape_search() (in module *robottelo.helpers*), 153
- execute() (*robottelo.cli.base.Base* class method), 37
- execute_command() (in module *robottelo.ssh*), 176
- execute_foreman_scap_client() (*robottelo.vm.VirtualMachine* method), 188
- exists() (*robottelo.cli.base.Base* class method), 38
- export() (*robottelo.cli.repository.Repository* class method), 93
- exports() (*robottelo.cli.template_sync.TemplateSync* class method), 105
- extract_capsule_satellite_installer_command() (in module *robottelo.helpers*), 156
- extract_min_version() (in module *robottelo.utils.issue_handlers.bugzilla*), 136
- extract_ui_token() (in module *robottelo.helpers*), 156
- ## F
- Fact (class in *robottelo.cli.fact*), 51
- facts() (*robottelo.cli.discoveredhost.DiscoveredHost* class method), 45
- facts() (*robottelo.cli.host.Host* class method), 69
- FAKE_0_CUSTOM_PACKAGE (in module *robottelo.constants*), 144
- FAKE_0_CUSTOM_PACKAGE_GROUP (in module *robottelo.constants*), 145
- FAKE_0_CUSTOM_PACKAGE_GROUP_NAME (in module *robottelo.constants*), 144
- FAKE_0_CUSTOM_PACKAGE_NAME (in module *robottelo.constants*), 144
- FAKE_0_ERRATA_ID (in module *robottelo.constants*), 144
- FAKE_0_INC_UPD_ERRATA (in module *robottelo.constants*), 144
- FAKE_0_INC_UPD_NEW_PACKAGE (in module *robottelo.constants*), 144
- FAKE_0_INC_UPD_NEW_UPDATEFILE (in module *robottelo.constants*), 144
- FAKE_0_INC_UPD_OLD_PACKAGE (in module *robottelo.constants*), 144
- FAKE_0_INC_UPD_OLD_UPDATEFILE (in module *robottelo.constants*), 144
- FAKE_0_INC_UPD_URL (in module *robottelo.constants*), 144
- FAKE_0_MODULAR_ERRATA_ID (in module *robottelo.constants*), 144
- FAKE_0_PUPPET_MODULE (in module *robottelo.constants*), 145
- FAKE_0_PUPPET_REPO (in module *robottelo.constants*), 143
- FAKE_0_YUM_ERRATUM_COUNT (in module *robottelo.constants*), 145
- FAKE_0_YUM_REPO (in module *robottelo.constants*), 143
- FAKE_0_YUM_REPO_PACKAGES_COUNT (in module *robottelo.constants*), 143
- FAKE_0_YUM_REPO_STRING_BASED_VERSIONS (in module *robottelo.constants*), 145
- FAKE_0_YUM_REPO_STRING_BASED_VERSIONS_COUNTS (in module *robottelo.constants*), 145
- FAKE_10_YUM_BUGFIX_ERRATUM (in module *robottelo.constants*), 145
- FAKE_10_YUM_BUGFIX_ERRATUM_COUNT (in module *robottelo.constants*), 145

FAKE_10_YUM_REPO (in module *robottelo.constants*), 143

FAKE_11_YUM_ENHANCEMENT_ERRATUM (in module *robottelo.constants*), 145

FAKE_11_YUM_ENHANCEMENT_ERRATUM_COUNT (in module *robottelo.constants*), 145

FAKE_11_YUM_REPO (in module *robottelo.constants*), 143

FAKE_1_CUSTOM_PACKAGE (in module *robottelo.constants*), 144

FAKE_1_CUSTOM_PACKAGE_NAME (in module *robottelo.constants*), 144

FAKE_1_ERRATA_ID (in module *robottelo.constants*), 144

FAKE_1_PUPPET_REPO (in module *robottelo.constants*), 143

FAKE_1_YUM_ERRATUM_COUNT (in module *robottelo.constants*), 145

FAKE_1_YUM_REPO (in module *robottelo.constants*), 143

FAKE_1_YUM_REPO_RPMS (in module *robottelo.constants*), 145

FAKE_1_YUM_REPOS_COUNT (in module *robottelo.constants*), 145

FAKE_2_CUSTOM_PACKAGE (in module *robottelo.constants*), 144

FAKE_2_CUSTOM_PACKAGE_NAME (in module *robottelo.constants*), 144

FAKE_2_ERRATA_ID (in module *robottelo.constants*), 144

FAKE_2_PUPPET_REPO (in module *robottelo.constants*), 143

FAKE_2_YUM_REPO (in module *robottelo.constants*), 143

FAKE_3_CUSTOM_PACKAGE (in module *robottelo.constants*), 144

FAKE_3_CUSTOM_PACKAGE_NAME (in module *robottelo.constants*), 144

FAKE_3_ERRATA_ID (in module *robottelo.constants*), 144

FAKE_3_PUPPET_REPO (in module *robottelo.constants*), 143

FAKE_3_YUM_ERRATUM_COUNT (in module *robottelo.constants*), 145

FAKE_3_YUM_REPO (in module *robottelo.constants*), 143

FAKE_3_YUM_REPOS_COUNT (in module *robottelo.constants*), 145

FAKE_4_CUSTOM_PACKAGE (in module *robottelo.constants*), 144

FAKE_4_CUSTOM_PACKAGE_NAME (in module *robottelo.constants*), 144

FAKE_4_ERRATA_ID (in module *robottelo.constants*), 145

FAKE_4_PUPPET_REPO (in module *robottelo.constants*), 143

FAKE_4_YUM_REPO (in module *robottelo.constants*), 143

FAKE_5_CUSTOM_PACKAGE (in module *robottelo.constants*), 144

FAKE_5_ERRATA_ID (in module *robottelo.constants*), 145

FAKE_5_PUPPET_REPO (in module *robottelo.constants*), 143

FAKE_5_YUM_REPO (in module *robottelo.constants*), 143

FAKE_6_PUPPET_REPO (in module *robottelo.constants*), 143

FAKE_6_YUM_ERRATUM_COUNT (in module *robottelo.constants*), 145

FAKE_6_YUM_REPO (in module *robottelo.constants*), 143

FAKE_7_PUPPET_REPO (in module *robottelo.constants*), 144

FAKE_7_YUM_REPO (in module *robottelo.constants*), 143

FAKE_8_PUPPET_REPO (in module *robottelo.constants*), 144

FAKE_8_YUM_REPO (in module *robottelo.constants*), 143

FAKE_9_YUM_ERRATUM (in module *robottelo.constants*), 145

FAKE_9_YUM_ERRATUM_COUNT (in module *robottelo.constants*), 145

FAKE_9_YUM_OUTDATED_PACKAGES (in module *robottelo.constants*), 144

FAKE_9_YUM_REPO (in module *robottelo.constants*), 143

FAKE_9_YUM_SECURITY_ERRATUM (in module *robottelo.constants*), 145

FAKE_9_YUM_SECURITY_ERRATUM_COUNT (in module *robottelo.constants*), 145

FAKE_9_YUM_UPDATED_PACKAGES (in module *robottelo.constants*), 144

FAKE_PULP_REMOTE_FILEREPO (in module *robottelo.constants*), 145

FAKE_YUM_DRPM_REPO (in module *robottelo.constants*), 143

FAKE_YUM_MIXED_REPO (in module *robottelo.constants*), 143

FAKE_YUM_SRPM_DUPLICATE_REPO (in module *robottelo.constants*), 143

FAKE_YUM_SRPM_REPO (in module *robottelo.constants*), 143

FakeCapsuleSettings (class in *robottelo.config.base*), 114

FakeManifestSettings (class in *robottelo.config.base*), 113

- FAM_MODULE_PATH (in module *robottelo.constants*), 147
- FeatureSettings (class in *robottelo.config.base*), 111
- FEDORA26_OSTREE_REPO (in module *robottelo.constants*), 144
- FEDORA27_OSTREE_REPO (in module *robottelo.constants*), 144
- fetch() (*robottelo.cli.virt_who_config.VirtWhoConfig* class method), 108
- File (class in *robottelo.cli.file*), 63
- file_downloader() (in module *robottelo.helpers*), 153
- FileStorageHandler (class in *robottelo.decorators.func_shared.file_storage*), 121
- Filter (class in *robottelo.cli.filter*), 64
- filter (*robottelo.cli.contentview.ContentView* attribute), 42
- filter() (*robottelo.log.LogFile* method), 159
- FILTER_CONTENT_TYPE (in module *robottelo.constants*), 142
- FILTER_ERRATA_DATE (in module *robottelo.constants*), 142
- FILTER_ERRATA_TYPE (in module *robottelo.constants*), 142
- FILTER_TYPE (in module *robottelo.constants*), 142
- filtered_datapoint() (in module *robottelo.datafactory*), 148
- filters() (*robottelo.cli.role.Role* class method), 95
- fixture (in module *robottelo.decorators*), 131
- follow_duplicates() (in module *robottelo.utils.issue_handlers.bugzilla*), 136
- FOREMAN_ANSIBLE_MODULES (in module *robottelo.constants*), 146
- FOREMAN_PROVIDERS (in module *robottelo.constants*), 140
- FOREMAN_TEMPLATE_IMPORT_URL (in module *robottelo.constants*), 146
- FOREMAN_TEMPLATE_ROOT_DIR (in module *robottelo.constants*), 146
- FOREMAN_TEMPLATE_TEST_TEMPLATE (in module *robottelo.constants*), 146
- FOREMAN_TEMPLATES_COMMUNITY_URL (in module *robottelo.constants*), 146
- form_repo_path() (in module *robottelo.helpers*), 155
- FunctionLockerError, 129
- G**
- GCE_EXTERNAL_IP_DEFAULT (in module *robottelo.constants*), 141
- GCE_MACHINE_TYPE_DEFAULT (in module *robottelo.constants*), 140
- GCE_NETWORK_DEFAULT (in module *robottelo.constants*), 140
- GCECertNotFoundError, 151
- GCESettings (class in *robottelo.config.base*), 113
- generate() (*robottelo.cli.report_template.ReportTemplate* class method), 92
- generate_strings_list() (in module *robottelo.datafactory*), 148
- generate_system_facts() (in module *robottelo.system_facts*), 177
- GenericRHRRepository (class in *robottelo.products*), 168
- get() (*robottelo.config.base.INIReader* method), 110
- get() (*robottelo.decorators.func_shared.base.BaseStorageHandler* method), 121
- get() (*robottelo.decorators.func_shared.file_storage.FileStorageHandler* method), 122
- get() (*robottelo.decorators.func_shared.redis_storage.RedisStorageHandler* method), 123
- get() (*robottelo.vm.VirtualMachine* method), 188
- get_available_capsule_port() (in module *robottelo.helpers*), 154
- get_cert_rpm_url() (*robottelo.config.base.ServerSettings* method), 112
- get_client() (in module *robottelo.ssh*), 174
- get_configure_command() (in module *robottelo.virtwho_utils*), 183
- get_configure_file() (in module *robottelo.virtwho_utils*), 183
- get_configure_id() (in module *robottelo.virtwho_utils*), 183
- get_configure_option() (in module *robottelo.virtwho_utils*), 183
- get_connection() (in module *robottelo.ssh*), 174
- get_credentials() (*robottelo.config.base.ServerSettings* method), 111
- get_data_bz() (in module *robottelo.utils.issue_handlers.bugzilla*), 137
- get_data_file() (in module *robottelo.helpers*), 154
- get_default_bz() (in module *robottelo.utils.issue_handlers.bugzilla*), 137
- get_func_name() (in module *robottelo.helpers*), 155
- get_guest_info() (in module *robottelo.virtwho_utils*), 182
- get_host_info() (in module *robottelo.helpers*), 153
- get_host_os_version() (in module *robottelo.host_info*), 157
- get_host_sat_version() (in module *robottelo.host_info*), 157
- get_hypervisor_info() (in module *robottelo.virtwho_utils*), 184
- get_key_file_path() (*robottelo.helpers* method), 155

- telo.decorators.func_shared.file_storage.FileStorage* (method), 122
- `get_line_indentation_level()` (in module *robottelo.cli.hammer*), 67
- `get_line_indentation_spaces()` (in module *robottelo.cli.hammer*), 66
- `get_nailgun_config()` (in module *robottelo.helpers*), 153
- `get_output()` (*robottelo.cli.job_invocation.JobInvocation* class method), 76
- `get_project_root()` (in module *robottelo.config.base*), 110
- `get_pub_url()` (*robottelo.config.base.ServerSettings* method), 111
- `get_repo_files()` (in module *robottelo.host_info*), 157
- `get_repomd_revision()` (in module *robottelo.host_info*), 158
- `get_rhsso_client_id()` (in module *robottelo.rhsso_utils*), 172
- `get_rhsso_user_details()` (in module *robottelo.rhsso_utils*), 172
- `get_sat_version()` (in module *robottelo.host_info*), 158
- `get_server_distro()` (in module *robottelo.products*), 166
- `get_server_software()` (in module *robottelo.helpers*), 153
- `get_server_version()` (in module *robottelo.helpers*), 153
- `get_services_status()` (in module *robottelo.helpers*), 155
- `get_sftp_session()` (in module *robottelo.ssh*), 174
- `get_single_bz()` (in module *robottelo.utils.issue_handlers.bugzilla*), 137
- `get_system()` (in module *robottelo.virtwho_utils*), 182
- `get_temp_dir()` (in module *robottelo.decorators.func_locker*), 129
- `get_temp_dir()` (in module *robottelo.decorators.func_shared.file_storage*), 121
- `get_url()` (*robottelo.config.base.ServerSettings* method), 111
- `get_virtwho_status()` (in module *robottelo.virtwho_utils*), 183
- `get_web_session()` (in module *robottelo.helpers*), 156
- GlobalParameter* (class in *robottelo.cli.globalparam*), 65
- GPGKey* (class in *robottelo.cli.gpgkey*), 65
- `GROUP_MEMBERSHIP_MAPPER` (in module *robottelo.constants*), 146
- Handler*
 - `HAMMER_CONFIG` (in module *robottelo.constants*), 146
 - `handler_methods` (in module *robottelo.utils.issue_handlers*), 137
 - `has_section()` (*robottelo.config.base.INIReader* method), 110
 - `HASH_TYPE` (in module *robottelo.constants*), 141
 - `headers` (*robottelo.report_portal.portal.ReportPortal* attribute), 134
 - Host* (class in *robottelo.cli.host*), 68
 - `host_cleanup()` (in module *robottelo.cleanup*), 139
 - `host_collection()` (*robottelo.cli.activationkey.ActivationKey* class method), 31
 - `host_location_update()` (in module *robottelo.upgrade_utility*), 181
 - `host_provisioning_check()` (in module *robottelo.helpers*), 156
 - `host_status()` (in module *robottelo.upgrade_utility*), 181
 - HostCollection* (class in *robottelo.cli.hostcollection*), 74
 - HostGroup* (class in *robottelo.cli.hostgroup*), 75
 - HostInfoError*, 152
 - HostInterface* (class in *robottelo.cli.host*), 73
 - `hostname` (*robottelo.config.base.CapsuleSettings* attribute), 112
 - `hostname` (*robottelo.vm.VirtualMachine* attribute), 185
 - `hostname_local` (*robottelo.vm_capsule.CapsuleVirtualMachine* attribute), 189
 - `hosts()` (*robottelo.cli.hostcollection.HostCollection* class method), 74
 - `HOT_BACKUP_FILES` (in module *robottelo.constants*), 146
 - `HTML_TAGS` (in module *robottelo.constants*), 141
 - HttpProxy* (class in *robottelo.cli.http_proxy*), 76
 - HttpProxySettings* (class in *robottelo.config.base*), 118
 - `hypervisor_json_create()` (in module *robottelo.virtwho_utils*), 184
- I**
 - `idgen()` (in module *robottelo.helpers*), 156
 - `image_available()` (*robottelo.cli.computeresource.ComputeResource* class method), 41
 - `image_create()` (*robottelo.cli.computeresource.ComputeResource* class method), 40
 - `image_delete()` (*robottelo.cli.computeresource.ComputeResource* class method), 41

- image_info() (*robot-telo.cli.computeresource.ComputeResource class method*), 40
- image_list() (*robot-telo.cli.computeresource.ComputeResource class method*), 41
- image_update() (*robot-telo.cli.computeresource.ComputeResource class method*), 41
- import_classes() (*robottelo.cli.capsule.Capsule class method*), 39
- import_classes() (*robottelo.cli.proxy.Proxy class method*), 89
- imports() (*robottelo.cli.template_sync.TemplateSync class method*), 105
- ImproperlyConfigured, 110
- info() (*robottelo.cli.base.Base class method*), 38
- info() (*robottelo.cli.docker.DockerManifest class method*), 46
- info() (*robottelo.cli.docker.DockerTag class method*), 47
- info() (*robottelo.cli.gpgkey.GPGKey class method*), 65
- info() (*robottelo.cli.repository.Repository class method*), 93
- info() (*robottelo.cli.sparams.SmartClassParameter class method*), 98
- info() (*robottelo.cli.srpm.Srpm class method*), 100
- INIReader (*class in robottelo.config.base*), 110
- install_capsule_katello_ca() (*robot-telo.vm.VirtualMachine method*), 186
- install_katello_agent() (*robot-telo.vm.VirtualMachine method*), 186
- install_katello_ca() (*in module robot-telo.helpers*), 154
- install_katello_ca() (*robot-telo.vm.VirtualMachine method*), 186
- install_katello_host_tools() (*robot-telo.vm.VirtualMachine method*), 186
- INSTALL_MEDIUM_URL (*in module robot-telo.constants*), 141
- install_or_update_package() (*in module robottelo.upgrade_utility*), 181
- INTERFACE_API (*in module robottelo.constants*), 140
- INTERFACE_CLI (*in module robottelo.constants*), 140
- invalid_boolean_strings() (*in module robot-telo.datafactory*), 148
- invalid_docker_upstream_names() (*in module robottelo.datafactory*), 150
- invalid_domain_names() (*in module robot-telo.datafactory*), 149
- invalid_emails_list() (*in module robot-telo.datafactory*), 148
- invalid_http_credentials() (*in module robot-telo.datafactory*), 150
- invalid_id_list() (*in module robot-telo.datafactory*), 149
- invalid_interfaces_list() (*in module robot-telo.datafactory*), 150
- invalid_names_list() (*in module robot-telo.datafactory*), 149
- INVALID_URL (*in module robottelo.constants*), 144
- invalid_usernames_list() (*in module robot-telo.datafactory*), 149
- invalid_values_list() (*in module robot-telo.datafactory*), 149
- InvalidArgumentError, 148, 152
- is_distro_repository (*robot-telo.products.GenericRHRRepository attribute*), 168
- is_open() (*in module robottelo.utils.issue_handlers*), 137
- is_open_bz() (*in module robot-telo.utils.issue_handlers.bugzilla*), 136
- is_ssh_pub_key() (*in module robottelo.ssh*), 176
- ## J
- JobInvocation (*class in robot-telo.cli.job_invocation*), 76
- JobTemplate (*class in robottelo.cli.job_template*), 77
- ## K
- key (*robot-telo.decorators.func_shared.shared._SharedFunction attribute*), 126
- key (*robot-telo.products.GenericRHRRepository attribute*), 168
- KEY_CLOAK_CLI (*in module robottelo.constants*), 141
- kinds() (*robot-telo.cli.template.Template class method*), 104
- ## L
- LANGUAGES (*in module robottelo.constants*), 142
- LATEST_RHEL7_GCE_IMG_UUID (*in module robot-telo.constants*), 140
- Launch (*class in robottelo.report_portal.portal*), 134
- launch() (*robot-telo.report_portal.portal.ReportPortal method*), 134
- launch_types (*in module robot-telo.report_portal.portal*), 133
- launches() (*robot-telo.report_portal.portal.ReportPortal method*), 134
- LDAP_ATTR (*in module robottelo.constants*), 146
- LDAP_SERVER_TYPE (*in module robottelo.constants*), 146
- LDAPAuthSource (*class in robot-telo.cli.ldapauthsource*), 77
- LDAPIPASettings (*class in robottelo.config.base*), 114
- LDAPSettings (*class in robottelo.config.base*), 114

- LIBVIRT_RESOURCE_URL (in module *robottelo.constants*), 140
 - LibvirtGuest (class in *robottelo.libvirt_discovery*), 158
 - LibvirtGuestError, 158
 - LibvirtHostSettings (class in *robottelo.config.base*), 114
 - LIFECYCLE_KEYS (in module *robottelo.cli.factory*), 53
 - LifecycleEnvironment (class in *robottelo.cli.lifecycleenvironment*), 78
 - List (class in *robottelo.config.casts*), 118
 - list () (*robottelo.cli.arfreport.Arfreport* class method), 33
 - list () (*robottelo.cli.base.Base* class method), 38
 - list () (*robottelo.cli.docker.DockerManifest* class method), 46
 - list () (*robottelo.cli.docker.DockerTag* class method), 48
 - list () (*robottelo.cli.lifecycleenvironment.LifecycleEnvironment* class method), 78
 - list () (*robottelo.cli.srpm.Srpm* class method), 100
 - list_tasks () (*robottelo.cli.task.Task* class method), 103
 - LOCALES (in module *robottelo.constants*), 140
 - Location (class in *robottelo.cli.location*), 79
 - location_cleanup () (in module *robottelo.cleanup*), 139
 - lock () (*robottelo.decorators.func_shared.base.BaseStorageHandler* class method), 121
 - lock () (*robottelo.decorators.func_shared.file_storage.FileStorageHandler* class method), 122
 - lock () (*robottelo.decorators.func_shared.redis_storage.RedisStorageHandler* class method), 122
 - LOCK_DEFAULT_SCOPE (in module *robottelo.decorators.func_locker*), 129
 - LOCK_DEFAULT_TIMEOUT (in module *robottelo.decorators.func_locker*), 129
 - LOCK_DIR (in module *robottelo.decorators.func_locker*), 129
 - LOCK_FILE_NAME_EXT (in module *robottelo.decorators.func_locker*), 129
 - lock_function () (in module *robottelo.decorators.func_locker*), 130
 - LOCK_TIMEOUT (in module *robottelo.decorators.func_shared.file_storage*), 121
 - LOCK_TIMEOUT (in module *robottelo.decorators.func_shared.redis_storage*), 122
 - locking_function () (in module *robottelo.decorators.func_locker*), 130
 - LogFile (class in *robottelo.log*), 159
 - LOGGER (in module *robottelo.cleanup*), 139
 - logger (in module *robottelo.cli.factory*), 53
 - LOGGER (in module *robottelo.config.base*), 110
 - LOGGER (in module *robottelo.decorators*), 131
 - logger (in module *robottelo.decorators.func_locker*), 129
 - logger (in module *robottelo.decorators.func_shared.file_storage*), 121
 - logger (in module *robottelo.decorators.func_shared.shared*), 125
 - LOGGER (in module *robottelo.decorators.host*), 130
 - LOGGER (in module *robottelo.helpers*), 152
 - LOGGER (in module *robottelo.host_info*), 157
 - logger (in module *robottelo.libvirt_discovery*), 158
 - LOGGER (in module *robottelo.report_portal.portal*), 133
 - logger (in module *robottelo.ssh*), 173
 - LOGGER (in module *robottelo.test*), 178
 - LOGGER (in module *robottelo.utils.issue_handlers.bugzilla*), 136
 - logger (in module *robottelo.vm*), 185
 - logger (in module *robottelo.vm_capsule*), 189
 - logger (*robottelo.cli.base.Base* attribute), 37
 - LoggingLevel (class in *robottelo.config.casts*), 119
 - login () (*robottelo.cli.auth.Auth* class method), 34
 - logout () (*robottelo.cli.auth.Auth* class method), 35
 - LOGS_DATA_DIR (in module *robottelo.log*), 159
- ## M
- MAJOR_VERSION_DISTRO (in module *robottelo.constants*), 140
 - make_authentication_key () (in module *robottelo.cli.factory*), 54
 - make_basearchitecture () (in module *robottelo.cli.factory*), 54
 - make_compute_resource () (in module *robottelo.cli.factory*), 57
 - make_content_view () (in module *robottelo.cli.factory*), 54
 - make_content_view_filter () (in module *robottelo.cli.factory*), 54
 - make_content_view_filter_rule () (in module *robottelo.cli.factory*), 54
 - make_content_view_with_credentials () (in module *robottelo.cli.factory*), 54
 - make_discoveryrule () (in module *robottelo.cli.factory*), 55
 - make_domain () (in module *robottelo.cli.factory*), 58
 - make_environment () (in module *robottelo.cli.factory*), 58
 - make_fake_host () (in module *robottelo.cli.factory*), 56
 - make_filter () (in module *robottelo.cli.factory*), 56
 - make_gpg_key () (in module *robottelo.cli.factory*), 55
 - make_host () (in module *robottelo.cli.factory*), 56

- make_host_collection()* (in module *robot-telo.cli.factory*), 56
make_hostgroup() (in module *robottelo.cli.factory*), 58
make_http_proxy() (in module *robot-telo.cli.factory*), 63
make_job_invocation() (in module *robot-telo.cli.factory*), 56
make_job_template() (in module *robot-telo.cli.factory*), 57
make_ldap_auth_source() (in module *robot-telo.cli.factory*), 57
make_lifecycle_environment() (in module *robottelo.cli.factory*), 58
make_location() (in module *robottelo.cli.factory*), 55
make_medium() (in module *robottelo.cli.factory*), 58
make_model() (in module *robottelo.cli.factory*), 55
make_org() (in module *robottelo.cli.factory*), 57
make_org_with_credentials() (in module *robottelo.cli.factory*), 57
make_os() (in module *robottelo.cli.factory*), 57
make_partition_table() (in module *robot-telo.cli.factory*), 55
make_product() (in module *robottelo.cli.factory*), 55
make_product_wait() (in module *robot-telo.cli.factory*), 55
make_product_with_credentials() (in module *robottelo.cli.factory*), 55
make_proxy() (in module *robottelo.cli.factory*), 55
make_realm() (in module *robottelo.cli.factory*), 57
make_report_template() (in module *robot-telo.cli.factory*), 57
make_repository() (in module *robot-telo.cli.factory*), 55
make_repository_with_credentials() (in module *robottelo.cli.factory*), 56
make_role() (in module *robottelo.cli.factory*), 56
make_scap_policy() (in module *robot-telo.cli.factory*), 56
make_scapcontent() (in module *robot-telo.cli.factory*), 58
make_subnet() (in module *robottelo.cli.factory*), 56
make_sync_plan() (in module *robottelo.cli.factory*), 56
make_tailoringfile() (in module *robot-telo.cli.factory*), 58
make_template() (in module *robottelo.cli.factory*), 58
make_template_input() (in module *robot-telo.cli.factory*), 58
make_user() (in module *robottelo.cli.factory*), 57
make_usergroup() (in module *robottelo.cli.factory*), 57
make_usergroup_external() (in module *robot-telo.cli.factory*), 57
make_virt_who_config() (in module *robot-telo.cli.factory*), 59
Manifest (class in *robottelo.manifests*), 160
manifest (*robottelo.cli.docker.Docker* attribute), 49
manifest_history() (*robot-telo.cli.subscription.Subscription* class method), 101
ManifestCloner (class in *robottelo.manifests*), 160
MAX_SHARE_TIMEOUT (*robot-telo.config.base.SharedFunctionSettings* attribute), 116
md5_by_url() (in module *robottelo.helpers*), 154
Medium (class in *robottelo.cli.medium*), 81
MEMORY_CAPACITY (in module *robot-telo.system_facts*), 177
MEMORY_SIZE (in module *robottelo.system_facts*), 177
Model (class in *robottelo.cli.model*), 81
ModuleStream (class in *robottelo.cli.module_stream*), 82
- ## N
- NAMESPACE_SCOPE (in module *robot-telo.decorators.func_shared.shared*), 125
need_subscription (*robot-telo.products.RepositoryCollection* attribute), 170
networks() (*robottelo.cli.computeresource.ComputeResource* class method), 41
NO_REPOS_AVAILABLE (in module *robot-telo.constants*), 142
NOT_IMPLEMENTED (in module *robottelo.constants*), 141
NOT_IMPLEMENTED (in module *robottelo.decorators*), 131
NotRaisesValueHandler (class in *robottelo.test*), 178
- ## O
- oauth()* (*robottelo.cli.auth.AuthLogin* class method), 35
OBJECT_CACHE (in module *robottelo.decorators*), 131
one_to_many_names() (in module *robot-telo.api.utils*), 26
one_to_one_names() (in module *robot-telo.api.utils*), 26
OnlyOneOSRepositoryAllowed, 166
OPEN_STATUSES (in module *robottelo.constants*), 146
OPERATING_SYSTEMS (in module *robot-telo.constants*), 141
OperatingSys (class in *robottelo.cli.operatingsys*), 83
Org (class in *robottelo.cli.org*), 84
org_cleanup() (in module *robottelo.cleanup*), 139

- ORG_KEYS (in module *robottelo.cli.factory*), 53
 - organization (robottelo.products.RepositoryCollection attribute), 170
 - organization_has_manifest() (robottelo.products.RepositoryCollection static method), 170
 - original() (robottelo.manifests.ManifestCloner method), 160
 - original_manifest() (in module *robottelo.manifests*), 161
 - os_repo (robottelo.products.RepositoryCollection attribute), 170
 - OS_TEMPLATE_DATA_FILE (in module *robottelo.constants*), 141
 - OSCAP_DEFAULT_CONTENT (in module *robottelo.constants*), 146
 - OSCAP_PERIOD (in module *robottelo.constants*), 146
 - OSCAP_PROFILE (in module *robottelo.constants*), 146
 - OSCAP_WEEKDAY (in module *robottelo.constants*), 146
 - OscapSettings (class in *robottelo.config.base*), 115
 - OSPSettings (class in *robottelo.config.base*), 115
 - OstreeBranch (class in *robottelo.cli.ostreebranch*), 86
 - OstreeRepository (class in *robottelo.products*), 167
 - OstreeSettings (class in *robottelo.config.base*), 115
- ## P
- Package (class in *robottelo.cli.package*), 86
 - package_group_install() (robottelo.cli.host.Host class method), 69
 - package_group_remove() (robottelo.cli.host.Host class method), 69
 - package_install() (robottelo.cli.host.Host class method), 69
 - package_install() (robottelo.cli.hostcollection.HostCollection class method), 74
 - package_list() (robottelo.cli.host.Host class method), 69
 - package_remove() (robottelo.cli.host.Host class method), 69
 - package_upgrade() (robottelo.cli.host.Host class method), 69
 - package_upgrade_all() (robottelo.cli.host.Host class method), 69
 - parametrize (in module *robottelo.decorators*), 131
 - parametrized() (in module *robottelo.datafactory*), 148
 - parse_csv() (in module *robottelo.cli.hammer*), 66
 - parse_help() (in module *robottelo.cli.hammer*), 66
 - parse_info() (in module *robottelo.cli.hammer*), 67
 - parse_json() (in module *robottelo.cli.hammer*), 66
 - PARTITION_SCRIPT_DATA_FILE (in module *robottelo.constants*), 141
 - PartitionTable (class in *robottelo.cli.partitiontable*), 87
 - patch_os_release_version() (robottelo.vm.VirtualMachine method), 188
 - paths() (robottelo.cli.lifecycleenvironment.LifecycleEnvironment class method), 78
 - PerformanceSettings (class in *robottelo.config.base*), 115
 - PERMISSIONS (in module *robottelo.constants*), 145
 - PERMISSIONS_UI (in module *robottelo.constants*), 146
 - PRDS (in module *robottelo.constants*), 142
 - Product (class in *robottelo.cli.product*), 88
 - product_content() (robottelo.cli.activationkey.ActivationKey class method), 31
 - PRODUCT_KEY_ANSIBLE_ENGINE (in module *robottelo.products*), 166
 - PRODUCT_KEY_CLOUD_FORMS_TOOLS (in module *robottelo.products*), 166
 - PRODUCT_KEY_RHEL (in module *robottelo.products*), 166
 - PRODUCT_KEY_SAT_CAPSULE (in module *robottelo.products*), 166
 - PRODUCT_KEY_SAT_TOOLS (in module *robottelo.products*), 166
 - PRODUCT_KEY_VIRT_AGENTS (in module *robottelo.products*), 166
 - progress() (robottelo.cli.task.Task class method), 103
 - ProjectModeError, 133
 - promote() (in module *robottelo.api.utils*), 25
 - provision() (robottelo.cli.discoveredhost.DiscoveredHost class method), 45
 - ProvisioningCheckError, 152
 - Proxy (class in *robottelo.cli.proxy*), 89
 - ProxyError, 152
 - publish() (robottelo.cli.contentview.ContentView class method), 42
 - publish_content_view() (in module *robottelo.upgrade_utility*), 181
 - publish_puppet_module() (in module *robottelo.api.utils*), 25
 - publish_puppet_module() (in module *robottelo.cli.factory*), 60
 - PULP_PUBLISHED_ISO_REPOS_PATH (in module *robottelo.constants*), 145
 - PULP_PUBLISHED_PUPPET_REPOS_PATH (in module *robottelo.constants*), 145
 - PULP_PUBLISHED_YUM_REPOS_PATH (in module *robottelo.constants*), 145
 - Puppet (class in *robottelo.cli.puppet*), 89
 - puppet_module_add() (robottelo.cli.puppet class method), 89

- telo.cli.contentview.ContentView class method*), 42
- PUPPET_MODULE_CUSTOM_FILE_NAME (in module *robottelo.constants*), 145
- PUPPET_MODULE_CUSTOM_NAME (in module *robottelo.constants*), 145
- puppet_module_list() (*robottelo.cli.contentview.ContentView class method*), 42
- PUPPET_MODULE_NTP_PUPPETLABS (in module *robottelo.constants*), 145
- puppet_module_remove() (*robottelo.cli.contentview.ContentView class method*), 42
- puppet_modules (*robottelo.products.PuppetRepository attribute*), 167
- puppetclasses() (*robottelo.cli.base.Base class method*), 38
- PuppetModule (class in *robottelo.cli.puppetmodule*), 90
- PuppetRepository (class in *robottelo.products*), 167
- puppetrun() (*robottelo.cli.host.Host class method*), 69
- put() (*robottelo.vm.VirtualMachine method*), 188
- ## R
- read() (*robottelo.config.base.AzureRMSettings method*), 113
- read() (*robottelo.config.base.BugzillaSettings method*), 112
- read() (*robottelo.config.base.CapsuleSettings method*), 112
- read() (*robottelo.config.base.CertsSettings method*), 112
- read() (*robottelo.config.base.ClientsSettings method*), 112
- read() (*robottelo.config.base.ContainerRepositorySettings method*), 113
- read() (*robottelo.config.base.DiscoveryISOSettings method*), 115
- read() (*robottelo.config.base.DistroSettings method*), 113
- read() (*robottelo.config.base.DockerSettings method*), 113
- read() (*robottelo.config.base.EC2Settings method*), 113
- read() (*robottelo.config.base.FakeCapsuleSettings method*), 114
- read() (*robottelo.config.base.FakeManifestSettings method*), 113
- read() (*robottelo.config.base.FeatureSettings method*), 111
- read() (*robottelo.config.base.GCESettings method*), 114
- read() (*robottelo.config.base.HttpProxySettings method*), 118
- read() (*robottelo.config.base.LDAPIPASettings method*), 114
- read() (*robottelo.config.base.LDAPSettings method*), 114
- read() (*robottelo.config.base.LibvirtHostSettings method*), 114
- read() (*robottelo.config.base.OscapSettings method*), 115
- read() (*robottelo.config.base.OSPSettings method*), 115
- read() (*robottelo.config.base.OstreeSettings method*), 115
- read() (*robottelo.config.base.PerformanceSettings method*), 116
- read() (*robottelo.config.base.ReportPortalSettings method*), 117
- read() (*robottelo.config.base.RHAISettings method*), 116
- read() (*robottelo.config.base.RHEVSettings method*), 115
- read() (*robottelo.config.base.RHSSOSettings method*), 114
- read() (*robottelo.config.base.ServerSettings method*), 111
- read() (*robottelo.config.base.SharedFunctionSettings method*), 116
- read() (*robottelo.config.base.SSHClientSettings method*), 116
- read() (*robottelo.config.base.UpgradeSettings method*), 116
- read() (*robottelo.config.base.VirtWhoSettings method*), 117
- read() (*robottelo.config.base.VlanNetworkSettings method*), 116
- read() (*robottelo.config.base.VmWareSettings method*), 115
- read_data_file() (in module *robottelo.helpers*), 154
- REAL_0_ERRATA_ID (in module *robottelo.constants*), 145
- REAL_0_RH_PACKAGE (in module *robottelo.constants*), 144
- REAL_1_ERRATA_ID (in module *robottelo.constants*), 145
- REAL_2_ERRATA_ID (in module *robottelo.constants*), 145
- REAL_4_ERRATA_CVES (in module *robottelo.constants*), 145
- REAL_4_ERRATA_DETAILS (in module *robottelo.constants*), 146

REAL_4_ERRATA_ID (in module <i>robottelo.constants</i>), 145	<i>telo.cli.operatingsys.OperatingSys</i> class method), 83
REAL_RHEL7_0_0_PACKAGE (in module <i>robottelo.constants</i>), 144	<i>remove_capsule_katello_ca()</i> (<i>robottelo.vm.VirtualMachine</i> method), 187
REAL_RHEL7_0_0_PACKAGE_NAME (in module <i>robottelo.constants</i>), 144	<i>remove_compute_resource()</i> (<i>robottelo.cli.location.Location</i> class method), 80
REAL_RHEL7_0_1_PACKAGE (in module <i>robottelo.constants</i>), 144	<i>remove_compute_resource()</i> (<i>robottelo.cli.org.Org</i> class method), 84
REAL_RHEL7_0_1_PACKAGE_FILENAME (in module <i>robottelo.constants</i>), 144	<i>remove_content()</i> (<i>robottelo.cli.repository.Repository</i> class method), 94
REAL_RHEL7_0_ERRATA_ID (in module <i>robottelo.constants</i>), 145	<i>remove_domain()</i> (<i>robottelo.cli.location.Location</i> class method), 80
REAL_RHEL7_1_ERRATA_ID (in module <i>robottelo.constants</i>), 145	<i>remove_domain()</i> (<i>robottelo.cli.org.Org</i> class method), 84
Realm (class in <i>robottelo.cli.realm</i>), 91	<i>remove_environment()</i> (<i>robottelo.cli.location.Location</i> class method), 80
realm (in module <i>robottelo.rhssso_utils</i>), 171	<i>remove_environment()</i> (<i>robottelo.cli.org.Org</i> class method), 84
realm_cleanup() (in module <i>robottelo.cleanup</i>), 139	<i>remove_from_environment()</i> (<i>robottelo.cli.contentview.ContentView</i> class method), 43
reboot() (<i>robottelo.cli.host.Host</i> class method), 69	<i>remove_host()</i> (<i>robottelo.cli.hostcollection.HostCollection</i> class method), 74
RecurringLogic (class in <i>robottelo.cli.recurring_logic</i>), 91	<i>remove_host_collection()</i> (<i>robottelo.cli.activationkey.ActivationKey</i> class method), 31
redis (in module <i>robottelo.decorators.func_shared.redis_storage</i>), 122	<i>remove_hostgroup()</i> (<i>robottelo.cli.location.Location</i> class method), 80
REDIS_DB (in module <i>robottelo.decorators.func_shared.redis_storage</i>), 122	<i>remove_hostgroup()</i> (<i>robottelo.cli.org.Org</i> class method), 85
REDIS_HOST (in module <i>robottelo.decorators.func_shared.redis_storage</i>), 122	<i>remove_katello_ca()</i> (in module <i>robottelo.helpers</i>), 154
REDIS_PASSWORD (in module <i>robottelo.decorators.func_shared.redis_storage</i>), 122	<i>remove_katello_ca()</i> (<i>robottelo.vm.VirtualMachine</i> method), 187
REDIS_PORT (in module <i>robottelo.decorators.func_shared.redis_storage</i>), 122	<i>remove_location()</i> (<i>robottelo.cli.org.Org</i> class method), 85
RedisStorageHandler (class in <i>robottelo.decorators.func_shared.redis_storage</i>), 122	<i>remove_matcher()</i> (<i>robottelo.cli.sparams.SmartClassParameter</i> class method), 98
refresh() (<i>robottelo.cli.usergroup.UserGroupExternal</i> class method), 108	<i>remove_medium()</i> (<i>robottelo.cli.location.Location</i> class method), 80
refresh_features() (<i>robottelo.cli.capsule.Capsule</i> class method), 39	<i>remove_medium()</i> (<i>robottelo.cli.org.Org</i> class method), 85
refresh_features() (<i>robottelo.cli.proxy.Proxy</i> class method), 89	<i>remove_operating_system()</i> (<i>robottelo.cli.base.Base</i> class method), 38
refresh_manifest() (<i>robottelo.cli.subscription.Subscription</i> class method), 101	<i>remove_operatingsystem()</i> (<i>robottelo.cli.template.Template</i> class method), 104
register_contenthost() (<i>robottelo.vm.VirtualMachine</i> method), 187	<i>remove_organization()</i> (<i>robottelo.cli.org.Org</i> class method), 85
register_system() (in module <i>robottelo.virtwho_utils</i>), 183	
remove() (<i>robottelo.cli.contentview.ContentView</i> class method), 43	
remove_architecture() (<i>robottelo.cli.architecture.Architecture</i> class method), 80	

- telo.cli.location.Location* class method), 80
- remove_provisioning_template()* (*robottelo.cli.location.Location* class method), 80
- remove_provisioning_template()* (*robottelo.cli.operatingsys.OperatingSys* class method), 83
- remove_provisioning_template()* (*robottelo.cli.org.Org* class method), 85
- remove_ptable()* (*robottelo.cli.operatingsys.OperatingSys* class method), 83
- remove_repository()* (*robottelo.cli.activationkey.ActivationKey* class method), 31
- remove_repository()* (*robottelo.cli.contentview.ContentView* class method), 43
- remove_role()* (*robottelo.cli.user.User* class method), 106
- remove_role()* (*robottelo.cli.usergroup.UserGroup* class method), 107
- remove_smart_proxy()* (*robottelo.cli.location.Location* class method), 80
- remove_smart_proxy()* (*robottelo.cli.org.Org* class method), 85
- remove_subnet()* (*robottelo.cli.location.Location* class method), 80
- remove_subnet()* (*robottelo.cli.org.Org* class method), 85
- remove_subscription()* (*robottelo.cli.activationkey.ActivationKey* class method), 31
- remove_sync_plan()* (*robottelo.cli.product.Product* class method), 88
- remove_user()* (*robottelo.cli.location.Location* class method), 80
- remove_user()* (*robottelo.cli.org.Org* class method), 85
- remove_user()* (*robottelo.cli.usergroup.UserGroup* class method), 107
- remove_user_group()* (*robottelo.cli.usergroup.UserGroup* class method), 107
- remove_version()* (*robottelo.cli.contentview.ContentView* class method), 43
- REP_TEM_APPLIED_ERRATA_INPUT (in module *robottelo.constants*), 142
- repo_add_updateinfo()* (in module *robottelo.helpers*), 156
- repo_config_required* (*robottelo.config.base.ContainerRepositorySettings* attribute), 112
- repo_data* (*robottelo.products.GenericRHRepository* attribute), 168
- REPO_DISCOVERY_URL (in module *robottelo.constants*), 144
- repo_info* (*robottelo.products.BaseRepository* attribute), 167
- REPO_TAB (in module *robottelo.constants*), 141
- REPO_TYPE (in module *robottelo.constants*), 141
- REPO_TYPE_DOCKER (in module *robottelo.products*), 166
- REPO_TYPE_OSTREE (in module *robottelo.products*), 166
- REPO_TYPE_PUPPET (in module *robottelo.products*), 166
- REPO_TYPE_YUM (in module *robottelo.products*), 166
- Report (class in *robottelo.cli.report*), 92
- report_data()* (*robottelo.cli.report_template.ReportTemplate* class method), 93
- REPORT_TEMPLATE_FILE (in module *robottelo.constants*), 142
- ReportPortal (class in *robottelo.report_portal.portal*), 133
- ReportPortalSettings (class in *robottelo.config.base*), 117
- reports()* (*robottelo.cli.host.Host* class method), 70
- ReportTemplate (class in *robottelo.cli.report_template*), 92
- REPOS (in module *robottelo.constants*), 142
- repos_data* (*robottelo.products.RepositoryCollection* attribute), 170
- repos_info* (*robottelo.products.RepositoryCollection* attribute), 170
- ReposContentSetupWasNotPerformed, 166
- REPOSET (in module *robottelo.constants*), 142
- Repository (class in *robottelo.cli.repository*), 93
- RepositoryAlreadyCreated, 166
- RepositoryAlreadyDefinedError, 166
- RepositoryCollection (class in *robottelo.products*), 169
- RepositoryDataNotFound, 166
- RepositorySet (class in *robottelo.cli.repository_set*), 94
- RESOURCE_DEFAULT (in module *robottelo.constants*), 141
- restart_virtwho_service()* (in module *robottelo.virtwho_utils*), 184
- resume()* (*robottelo.cli.task.Task* class method), 103
- resume()* (*robottelo.vm_capsule.CapsuleVirtualMachine* method), 190
- rh_repos* (*robottelo.products.RepositoryCollection* attribute), 170
- rh_repos_info* (*robottelo.products.RepositoryCollection* attribute), 170

- telo.products.RepositoryCollection* attribute), 170
- rh_repository_id (*robottelo.products.GenericRHRepository* attribute), 168
- RHAISettings (class in *robottelo.config.base*), 116
- RHEL_6_MAJOR_VERSION (in module *robottelo.constants*), 140
- RHEL_7_MAJOR_VERSION (in module *robottelo.constants*), 140
- RHEL_8_MAJOR_VERSION (in module *robottelo.constants*), 140
- RHELAnsibleEngineRepository (class in *robottelo.products*), 169
- RHELCloudFormsTools (class in *robottelo.products*), 169
- RHELRepository (class in *robottelo.products*), 168
- RHEV_CR (in module *robottelo.constants*), 140
- RHEVSettings (class in *robottelo.config.base*), 114
- rhsso_host (in module *robottelo.rhsso_utils*), 171
- RHSSO_NEW_USER (in module *robottelo.constants*), 146
- rhsso_password (in module *robottelo.rhsso_utils*), 172
- RHSSO_RESET_PASSWORD (in module *robottelo.constants*), 146
- rhsso_user (in module *robottelo.rhsso_utils*), 172
- RHSSOSettings (class in *robottelo.config.base*), 114
- RHVA_REPO_TREE (in module *robottelo.constants*), 142
- robottelo (module), 22
- robottelo.api (module), 22
- robottelo.api.assertions (module), 22
- robottelo.api.utils (module), 23
- robottelo.cleanup (module), 139
- robottelo.cli (module), 30
- robottelo.cli.activationkey (module), 30
- robottelo.cli.ansible (module), 31
- robottelo.cli.architecture (module), 32
- robottelo.cli.arfreport (module), 32
- robottelo.cli.assertions (module), 33
- robottelo.cli.auth (module), 34
- robottelo.cli.base (module), 35
- robottelo.cli.capsule (module), 38
- robottelo.cli.computeprofile (module), 39
- robottelo.cli.computeresource (module), 40
- robottelo.cli.contentview (module), 41
- robottelo.cli.defaults (module), 43
- robottelo.cli.discoveredhost (module), 44
- robottelo.cli.discoveryrule (module), 45
- robottelo.cli.docker (module), 45
- robottelo.cli.domain (module), 49
- robottelo.cli.environment (module), 50
- robottelo.cli.erratum (module), 50
- robottelo.cli.fact (module), 51
- robottelo.cli.factory (module), 51
- robottelo.cli.file (module), 63
- robottelo.cli.filter (module), 63
- robottelo.cli.globalparam (module), 64
- robottelo.cli.gpgkey (module), 65
- robottelo.cli.hammer (module), 65
- robottelo.cli.host (module), 67
- robottelo.cli.hostcollection (module), 73
- robottelo.cli.hostgroup (module), 74
- robottelo.cli.http_proxy (module), 75
- robottelo.cli.job_invocation (module), 76
- robottelo.cli.job_template (module), 76
- robottelo.cli.ldapauthsource (module), 77
- robottelo.cli.lifecycleenvironment (module), 78
- robottelo.cli.location (module), 78
- robottelo.cli.medium (module), 80
- robottelo.cli.model (module), 81
- robottelo.cli.module_stream (module), 82
- robottelo.cli.operatingsys (module), 82
- robottelo.cli.org (module), 83
- robottelo.cli.ostreebranch (module), 85
- robottelo.cli.package (module), 86
- robottelo.cli.partitionable (module), 86
- robottelo.cli.product (module), 87
- robottelo.cli.proxy (module), 88
- robottelo.cli.puppet (module), 89
- robottelo.cli.puppetmodule (module), 90
- robottelo.cli.realm (module), 90
- robottelo.cli.recurring_logic (module), 91
- robottelo.cli.report (module), 91
- robottelo.cli.report_template (module), 92
- robottelo.cli.repository (module), 93
- robottelo.cli.repository_set (module), 94
- robottelo.cli.role (module), 95
- robottelo.cli.scap_policy (module), 96
- robottelo.cli.scap_tailoring_files (module), 96
- robottelo.cli.scapcontent (module), 97
- robottelo.cli.scpayloads (module), 98
- robottelo.cli.settings (module), 99
- robottelo.cli.srpm (module), 99
- robottelo.cli.subnet (module), 100
- robottelo.cli.subscription (module), 101
- robottelo.cli.syncplan (module), 101
- robottelo.cli.task (module), 102
- robottelo.cli.template (module), 103
- robottelo.cli.template_input (module), 104
- robottelo.cli.template_sync (module), 104
- robottelo.cli.user (module), 105
- robottelo.cli.usergroup (module), 106
- robottelo.cli.virt_who_config (module), 108
- robottelo.config (module), 109
- robottelo.config.base (module), 109

- robottelo.config.casts (*module*), 118
 - robottelo.constants (*module*), 140
 - robottelo.datafactory (*module*), 147
 - robottelo.decorators (*module*), 120
 - robottelo.decorators.func_locker (*module*), 127
 - robottelo.decorators.func_shared (*module*), 120
 - robottelo.decorators.func_shared.base (*module*), 120
 - robottelo.decorators.func_shared.file_storage (*module*), 121
 - robottelo.decorators.func_shared.redis_storage (*module*), 122
 - robottelo.decorators.func_shared.shared (*module*), 123
 - robottelo.decorators.host (*module*), 130
 - robottelo.errors (*module*), 151
 - robottelo.helpers (*module*), 151
 - robottelo.host_info (*module*), 157
 - robottelo.libvirt_discovery (*module*), 158
 - robottelo.log (*module*), 159
 - robottelo.manifests (*module*), 159
 - robottelo.products (*module*), 162
 - robottelo.report_portal (*module*), 133
 - robottelo.report_portal.portal (*module*), 133
 - robottelo.rhssso_utils (*module*), 171
 - robottelo.ssh (*module*), 172
 - robottelo.system_facts (*module*), 177
 - robottelo.test (*module*), 177
 - robottelo.ui (*module*), 135
 - robottelo.ui.utils (*module*), 135
 - robottelo.upgrade_utility (*module*), 180
 - robottelo.utils (*module*), 135
 - robottelo.utils.issue_handlers (*module*), 136
 - robottelo.utils.issue_handlers.bugzilla (*module*), 136
 - robottelo.utils.version (*module*), 138
 - robottelo.virtwho_utils (*module*), 181
 - robottelo.vm (*module*), 184
 - robottelo.vm_capsule (*module*), 188
 - Role (*class in robottelo.cli.role*), 95
 - ROLES (*in module robottelo.constants*), 146
 - roles_import () (*robottelo.cli.ansible.Ansible class method*), 31
 - roles_list () (*robottelo.cli.ansible.Ansible class method*), 31
 - ROLES_LOCKED (*in module robottelo.constants*), 146
 - ROLES_UNLOCKED (*in module robottelo.constants*), 146
 - root_dir (*robottelo.decorators.func_shared.file_storage.FileStorageHandler attribute*), 122
 - RPM_TO_UPLOAD (*in module robottelo.constants*), 141
 - rule (*robottelo.cli.contentview.ContentViewFilter attribute*), 42
 - run () (*robottelo.ssh.SSHClient method*), 173
 - run () (*robottelo.vm.VirtualMachine method*), 187
 - run_command () (*in module robottelo.rhssso_utils*), 172
 - run_goferd () (*in module robottelo.upgrade_utility*), 181
 - run_in_one_thread (*in module robottelo.decorators*), 131
 - run_only_on () (*in module robottelo.decorators*), 133
 - run_command () (*in module robottelo.virtwho_utils*), 182
- ## S
- Storage
 - SAT6_TOOLS_TREE (*in module robottelo.constants*), 142
 - satellite (*in module robottelo.rhssso_utils*), 171
 - SATELLITE_FIREWALL_SERVICE_NAME (*in module robottelo.constants*), 142
 - SATELLITE_SUBSCRIPTION_NAME (*in module robottelo.constants*), 142
 - SatelliteCapsuleRepository (*class in robottelo.products*), 169
 - SatelliteToolsRepository (*class in robottelo.products*), 169
 - SatVersionDependentValues (*class in robottelo.host_info*), 158
 - sc_params () (*robottelo.cli.base.Base class method*), 38
 - sc_params () (*robottelo.cli.environment.Environment class method*), 50
 - sc_params () (*robottelo.cli.host.Host class method*), 72
 - sc_params () (*robottelo.cli.hostgroup.HostGroup class method*), 75
 - sc_params () (*robottelo.cli.puppet.Puppet class method*), 89
 - Scapcontent (*class in robottelo.cli.scapcontent*), 97
 - Scappolicy (*class in robottelo.cli.scap_policy*), 96
 - schedule () (*robottelo.cli.report_template.ReportTemplate class method*), 93
 - search_version_key () (*in module robottelo.utils.version*), 138
 - section (*robottelo.config.base.ContainerRepositorySettings attribute*), 112
 - ServerFileDownloader (*class in robottelo.helpers*), 152
 - ServerSettings (*class in robottelo.config.base*), 111
 - set () (*robottelo.cli.globalparam.GlobalParameter class method*), 65
 - set () (*robottelo.cli.settings.Settings class method*), 99
 - set () (*robottelo.decorators.func_shared.base.BaseStorageHandler class method*), 121
 - set () (*robottelo.decorators.func_shared.file_storage.FileStorageHandler class method*), 122

set () (*robottelo.decorators.func_shared.redis_storage.RedisStorage* class method), 123
 set_default_scope () (in module *robottelo.decorators.func_locker*), 129
 set_default_scope () (in module *robottelo.decorators.func_shared.shared*), 125
 set_hammer_api_timeout () (in module *robottelo.api.utils*), 29
 set_infrastructure_type () (*robottelo.vm.VirtualMachine* method), 188
 set_parameter () (*robottelo.cli.base.Base* class method), 38
 set_sync_plan () (*robottelo.cli.product.Product* class method), 88
 setting_cleanup () (in module *robottelo.cleanup*), 139
 setting_is_set () (in module *robottelo.decorators*), 132
 Settings (class in *robottelo.cli.settings*), 99
 Settings (class in *robottelo.config*), 119
 Settings (class in *robottelo.config.base*), 117
 settings (in module *robottelo.config*), 120
 settings (in module *robottelo.decorators*), 131
 SETTINGS_FILE_NAME (in module *robottelo.config.base*), 110
 setup () (*robottelo.products.RepositoryCollection* method), 170
 setup_activation_key () (*robottelo.products.RepositoryCollection* static method), 170
 setup_cdn_and_custom_repos_content () (in module *robottelo.cli.factory*), 61
 setup_cdn_and_custom_repositories () (in module *robottelo.cli.factory*), 61
 setup_content () (*robottelo.products.RepositoryCollection* method), 170
 setup_content_data (*robottelo.products.RepositoryCollection* attribute), 170
 setup_content_view () (*robottelo.products.RepositoryCollection* method), 170
 setup_org_for_a_custom_repo () (in module *robottelo.cli.factory*), 59
 setup_org_for_a_rh_repo () (in module *robottelo.cli.factory*), 59
 setup_virtual_machine () (in module *robottelo.cli.factory*), 60
 setup_virtual_machine () (*robottelo.products.RepositoryCollection* method), 171
 setUpClass () (*robottelo.test.TestCase* class method), 179
 SSHClient (class in *robottelo.ssh*), 173
 SSHClientSettings (class in *robottelo.config.base*), 116
 SSHCommandResult (class in *robottelo.ssh*), 173
 SSHCommandTimeoutError, 173
 start () (*robottelo.cli.host.Host* class method), 70
 status () (*robottelo.cli.auth.Auth* class method), 35
 status () (*robottelo.cli.host.Host* class method), 70
 statuses (*robottelo.report_portal.portal.ReportPortal* attribute), 134
 stop () (*robottelo.cli.host.Host* class method), 70
 Storage (class in *robottelo.helpers*), 155
 STOP_TIMEOUT (in module *robottelo.decorators.func_shared.shared*), 125
 shared () (in module *robottelo.decorators.func_shared.shared*), 126
 SHARED_DIR (in module *robottelo.decorators.func_shared.file_storage*), 121
 SharedFunctionError, 126, 127
 SharedFunctionException, 126, 127
 SharedFunctionSettings (class in *robottelo.config.base*), 116
 should_deselect () (in module *robottelo.utils.issue_handlers*), 137
 should_deselect_bz () (in module *robottelo.utils.issue_handlers.bugzilla*), 136
 skip_if () (in module *robottelo.decorators*), 132
 skip_if_not_set () (in module *robottelo.decorators*), 132
 skip_if_os () (in module *robottelo.decorators.host*), 130
 skip_yum_update_during_provisioning () (in module *robottelo.api.utils*), 29
 skipif (in module *robottelo.decorators*), 131
 slugify_component () (in module *robottelo.helpers*), 156
 SM_OVERALL_STATUS (in module *robottelo.constants*), 142
 SmartClassParameter (class in *robottelo.cli.sparams*), 98
 SNIPPET_DATA_FILE (in module *robottelo.constants*), 141
 SNIPPET_URL (in module *robottelo.constants*), 141
 Srpm (class in *robottelo.cli.srpm*), 100
 SRPM_TO_UPLOAD (in module *robottelo.constants*), 141
 ssh_keys_add () (*robottelo.cli.user.User* class method), 106
 ssh_keys_delete () (*robottelo.cli.user.User* class method), 106
 ssh_keys_info () (*robottelo.cli.user.User* class method), 106
 ssh_keys_list () (*robottelo.cli.user.User* class method), 106
 SSHClient (class in *robottelo.ssh*), 173
 SSHClientSettings (class in *robottelo.config.base*), 116
 SSHCommandResult (class in *robottelo.ssh*), 173
 SSHCommandTimeoutError, 173
 start () (*robottelo.cli.host.Host* class method), 70
 status () (*robottelo.cli.auth.Auth* class method), 35
 status () (*robottelo.cli.host.Host* class method), 70
 statuses (*robottelo.report_portal.portal.ReportPortal* attribute), 134
 stop () (*robottelo.cli.host.Host* class method), 70
 Storage (class in *robottelo.helpers*), 155

- storage (*robottelo.decorators.func_shared.shared._SharedFunction attribute*), 126
- STRING_TYPES (*in module robottelo.constants*), 146
- stubbed() (*in module robottelo.decorators*), 132
- Subnet (*class in robottelo.cli.subnet*), 100
- SUBNET_IPAM_TYPES (*in module robottelo.constants*), 146
- subscribed (*robottelo.vm.VirtualMachine attribute*), 185
- Subscription (*class in robottelo.cli.subscription*), 101
- subscription_attach() (*robottelo.cli.host.Host class method*), 72
- subscription_auto_attach() (*robottelo.cli.host.Host class method*), 72
- subscription_manager_list_repos() (*robottelo.vm.VirtualMachine method*), 186
- subscription_manager_status() (*robottelo.vm.VirtualMachine method*), 186
- subscription_register() (*robottelo.cli.host.Host class method*), 71
- subscription_remove() (*robottelo.cli.host.Host class method*), 72
- subscription_unregister() (*robottelo.cli.host.Host class method*), 71
- subscriptions() (*robottelo.cli.activationkey.ActivationKey class method*), 31
- SUPPORTED_HANDLERS (*in module robottelo.utils.issue_handlers*), 137
- suspend() (*robottelo.vm_capsule.CapsuleVirtualMachine method*), 189
- SYNC_INTERVAL (*in module robottelo.constants*), 141
- synchronize() (*robottelo.cli.product.Product class method*), 88
- synchronize() (*robottelo.cli.repository.Repository class method*), 93
- synchronize() (*robottelo.products.BaseRepository method*), 167
- SyncPlan (*class in robottelo.cli.syncplan*), 102
- SYSTEM_FACTS (*in module robottelo.system_facts*), 177
- ## T
- tag (*robottelo.cli.docker.Docker attribute*), 49
- TailoringFiles (*class in robottelo.cli.scap_tailoring_files*), 97
- target_image (*robottelo.vm.VirtualMachine attribute*), 185
- Task (*class in robottelo.cli.task*), 102
- tearDownClass() (*robottelo.test.TestCase class method*), 179
- TEMP_FUNC_LOCK_DIR (*in module robottelo.decorators.func_locker*), 129
- TEMP_FUNC_SHARED_DIR (*in module robottelo.decorators.func_shared.file_storage*), 121
- TEMP_ROOT_DIR (*in module robottelo.decorators.func_locker*), 129
- TEMP_ROOT_DIR (*in module robottelo.decorators.func_shared.file_storage*), 121
- Template (*class in robottelo.cli.template*), 103
- TEMPLATE_TYPES (*in module robottelo.constants*), 141
- TemplateInput (*class in robottelo.cli.template_input*), 104
- TemplateNotFoundError, 151
- TemplateSync (*class in robottelo.cli.template_sync*), 105
- templateupdate (*class in robottelo.api.utils*), 28
- TestCase (*class in robottelo.test*), 179
- tests() (*robottelo.report_portal.portal.Launch method*), 135
- tier1 (*in module robottelo.decorators*), 131
- tier2 (*in module robottelo.decorators*), 131
- tier3 (*in module robottelo.decorators*), 131
- tier4 (*in module robottelo.decorators*), 131
- TIMEZONES (*in module robottelo.constants*), 142
- TOOLS_ERRATA_DETAILS (*in module robottelo.constants*), 146
- TOOLS_ERRATA_TABLE_DETAILS (*in module robottelo.constants*), 146
- transaction (*robottelo.decorators.func_shared.shared._SharedFunction attribute*), 126
- TREND_TYPES (*in module robottelo.constants*), 146
- try_from_cache() (*in module robottelo.utils.issue_handlers.bugzilla*), 136
- Tuple (*class in robottelo.config.casts*), 119
- ## U
- unregister() (*robottelo.vm.VirtualMachine method*), 187
- update() (*robottelo.cli.base.Base class method*), 38
- update_configure_option() (*in module robottelo.virtwho_utils*), 184
- update_dictionary() (*in module robottelo.helpers*), 154
- update_provisioning_template() (*in module robottelo.api.utils*), 29
- update_proxy() (*robottelo.cli.product.Product class method*), 88
- update_vm_host_location() (*in module robottelo.api.utils*), 28
- upgrade (*in module robottelo.decorators*), 131
- UpgradeSettings (*class in robottelo.config.base*), 116

- upload() (*robottelo.cli.subscription.Subscription class method*), 101
 - upload_content() (*robottelo.cli.repository.Repository class method*), 94
 - upload_file() (*in module robottelo.ssh*), 175
 - upload_files() (*in module robottelo.ssh*), 175
 - upload_manifest() (*in module robottelo.api.utils*), 25
 - upload_manifest() (*robottelo.test.TestCase class method*), 179
 - upload_manifest_locked() (*in module robottelo.manifests*), 161
 - upload_rhsso_entity() (*in module robottelo.rhssso_utils*), 172
 - upstream_name (*robottelo.products.DockerRepository attribute*), 167
 - url (*robottelo.products.BaseRepository attribute*), 167
 - url (*robottelo.products.GenericRHRepository attribute*), 168
 - url (*robottelo.products.RHELRepository attribute*), 169
 - url (*robottelo.products.SatelliteCapsuleRepository attribute*), 169
 - url (*robottelo.products.SatelliteToolsRepository attribute*), 169
 - User (*class in robottelo.cli.user*), 106
 - UserGroup (*class in robottelo.cli.usergroup*), 107
 - UserGroupExternal (*class in robottelo.cli.usergroup*), 107
- ## V
- valid_cron_expressions() (*in module robottelo.datafactory*), 150
 - valid_data_list() (*in module robottelo.datafactory*), 149
 - valid_docker_repository_names() (*in module robottelo.datafactory*), 149
 - valid_docker_upstream_names() (*in module robottelo.datafactory*), 150
 - valid_domain_names() (*in module robottelo.datafactory*), 149
 - valid_emails_list() (*in module robottelo.datafactory*), 149
 - valid_environments_list() (*in module robottelo.datafactory*), 149
 - VALID_GCE_ZONES (*in module robottelo.constants*), 140
 - VALID_GPG_KEY_BETA_FILE (*in module robottelo.constants*), 141
 - VALID_GPG_KEY_FILE (*in module robottelo.constants*), 141
 - valid_hostgroups_list() (*in module robottelo.datafactory*), 150
 - valid_hosts_list() (*in module robottelo.datafactory*), 149
 - valid_http_credentials() (*in module robottelo.datafactory*), 150
 - valid_interfaces_list() (*in module robottelo.datafactory*), 150
 - valid_labels_list() (*in module robottelo.datafactory*), 150
 - valid_names_list() (*in module robottelo.datafactory*), 150
 - valid_org_names_list() (*in module robottelo.datafactory*), 150
 - valid_url_list() (*in module robottelo.datafactory*), 150
 - valid_usernames_list() (*in module robottelo.datafactory*), 150
 - validate() (*robottelo.config.base.AzureRMSettings method*), 113
 - validate() (*robottelo.config.base.BugzillaSettings method*), 112
 - validate() (*robottelo.config.base.CapsuleSettings method*), 112
 - validate() (*robottelo.config.base.CertsSettings method*), 112
 - validate() (*robottelo.config.base.ClientsSettings method*), 112
 - validate() (*robottelo.config.base.ContainerRepositorySettings method*), 113
 - validate() (*robottelo.config.base.DiscoveryISOSettings method*), 115
 - validate() (*robottelo.config.base.DistroSettings method*), 113
 - validate() (*robottelo.config.base.DockerSettings method*), 113
 - validate() (*robottelo.config.base.EC2Settings method*), 113
 - validate() (*robottelo.config.base.FakeCapsuleSettings method*), 114
 - validate() (*robottelo.config.base.FakeManifestSettings method*), 113
 - validate() (*robottelo.config.base.FeatureSettings method*), 111
 - validate() (*robottelo.config.base.GCESettings method*), 114
 - validate() (*robottelo.config.base.HttpProxySettings method*), 118
 - validate() (*robottelo.config.base.LDAPIPASettings method*), 114
 - validate() (*robottelo.config.base.LDAPSettings method*), 114
 - validate() (*robottelo.config.base.LibvirtHostSettings method*), 114
 - validate() (*robottelo.config.base.OscapSettings method*), 115

- `validate()` (*robottelo.config.base.OSPSettings method*), 115
- `validate()` (*robottelo.config.base.OstreeSettings method*), 115
- `validate()` (*robottelo.config.base.PerformanceSettings method*), 116
- `validate()` (*robottelo.config.base.ReportPortalSettings method*), 117
- `validate()` (*robottelo.config.base.RHAISettings method*), 116
- `validate()` (*robottelo.config.base.RHEVSettings method*), 115
- `validate()` (*robottelo.config.base.RHSSOSettings method*), 114
- `validate()` (*robottelo.config.base.ServerSettings method*), 111
- `validate()` (*robottelo.config.base.SharedFunctionSettings method*), 117
- `validate()` (*robottelo.config.base.SSHClientSettings method*), 116
- `validate()` (*robottelo.config.base.UpgradeSettings method*), 116
- `validate()` (*robottelo.config.base.VirtWhoSettings method*), 117
- `validate()` (*robottelo.config.base.VlanNetworkSettings method*), 116
- `validate()` (*robottelo.config.base.VmWareSettings method*), 115
- `validate()` (*robottelo.test.APINotRaisesValueHandler method*), 178
- `validate()` (*robottelo.test.CLINotRaisesValueHandler method*), 178
- `validate()` (*robottelo.test.NotRaisesValueHandler method*), 178
- `value_name` (*robottelo.test.APINotRaisesValueHandler attribute*), 178
- `value_name` (*robottelo.test.CLINotRaisesValueHandler attribute*), 178
- `value_name` (*robottelo.test.NotRaisesValueHandler attribute*), 178
- `values_create()` (*robottelo.cli.computeprofile.ComputeProfile class method*), 40
- `variables_import()` (*robottelo.cli.ansible.Ansible class method*), 31
- `VDC_SUBSCRIPTION_NAME` (*in module robottelo.constants*), 142
- `version` (*robottelo.config.base.ServerSettings attribute*), 111
- `version_delete()` (*robottelo.cli.contentview.ContentView class method*), 43
- `version_export()` (*robottelo.cli.contentview.ContentView class method*), 43
- `version_import()` (*robottelo.cli.contentview.ContentView class method*), 43
- `version_incremental_update()` (*robottelo.cli.contentview.ContentView class method*), 42
- `version_info()` (*robottelo.cli.contentview.ContentView class method*), 42
- `version_list()` (*robottelo.cli.contentview.ContentView class method*), 42
- `version_promote()` (*robottelo.cli.contentview.ContentView class method*), 42
- `VERSION_RE` (*in module robottelo.utils.issue_handlers.bugzilla*), 136
- `VersionEncoder` (*class in robottelo.utils.version*), 138
- `virt_who_hypervisor_config()` (*in module robottelo.cli.factory*), 62
- `VIRT_WHO_HYPERVISOR_TYPES` (*in module robottelo.constants*), 140
- `VirtualizationAgentsRepository` (*class in robottelo.products*), 169
- `VirtualMachine` (*class in robottelo.vm*), 185
- `VirtualMachineError`, 185
- `virtwho_cleanup()` (*in module robottelo.virtwho_utils*), 183
- `virtwho_package_locked()` (*in module robottelo.virtwho_utils*), 184
- `VIRTWHO_SYSCONFIG` (*in module robottelo.virtwho_utils*), 182
- `VirtWhoConfig` (*class in robottelo.cli.virt_who_config*), 108
- `VirtWhoError`, 182
- `VirtWhoSettings` (*class in robottelo.config.base*), 117
- `VlanNetworkSettings` (*class in robottelo.config.base*), 116
- `vm_cleanup()` (*in module robottelo.cleanup*), 139
- `vm_setup_ssh_config()` (*in module robottelo.cli.factory*), 62
- `vm_upload_ssh_key()` (*in module robottelo.cli.factory*), 62
- `VMWARE_CONSTANTS` (*in module robottelo.constants*), 146
- `VmWareSettings` (*class in robottelo.config.base*), 115

W

- `wait_for_errata_applicability_task()` (*in module robottelo.api.utils*), 27

`wait_for_syncplan_tasks()` (in module `robottelo.api.utils`), 27

`wait_for_tasks()` (in module `robottelo.api.utils`), 27

`WebdriverDesiredCapabilities` (class in `robottelo.config.casts`), 119

`when_lock_acquired()` (`robottelo.decorators.func_shared.base.BaseStorageHandler` method), 121

`when_lock_acquired()` (`robottelo.decorators.func_shared.file_storage.FileStorageHandler` method), 122

`when_lock_acquired()` (`robottelo.decorators.func_shared.redis_storage.RedisStorageHandler` method), 122

`with_user()` (`robottelo.cli.base.Base` class method), 38

`WONTFIX_RESOLUTIONS` (in module `robottelo.constants`), 146

X

`xdist_adapter()` (in module `robottelo.datafactory`), 148

Y

`YumRepository` (class in `robottelo.products`), 167

Z

`ZOO_CUSTOM_GPG_KEY` (in module `robottelo.constants`), 141