Robottelo Documentation

Release 0.0.1

Og Maciel <omaciel@redhat.com>

Jan 16, 2023

Contents

1 Quickstart **3**
1.1 Requirements . . . 3

2 Running the Tests **5**
2.1 Initial Configuration . . . 5
2.2 Testing With Unittest . . . 6
2.3 Testing With Nose . . . 6
2.4 Running UI Tests On a Headless Server . . . 7

3 API Reference **9**
3.1 API Documentation . . . 9

4 Miscellany **21**

Python Module Index **23**

Index **25**

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.

CHAPTER 1

Quickstart

The following is only a brief setup guide for Robottelo. More extensive documentation is available at Read the Docs.

1.1 Requirements

Install Python header files. The package providing these files varies per distribution. For example:

- Fedora 20 provides header files in the python-devel package.
- Ubuntu 14.04 provides header files in the python-dev package.

Get the source code and install dependencies:

```
$ git clone git://github.com/omaciel/robottelo.git
$ pip install -r requirements.txt
```

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"`.
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 pylint.

CHAPTER 2

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
```

If you want to run the test suite without the aid of `make`, you can do that with either unittest or nose:

```
$ python -m unittest discover -s tests -t .
$ nosetests -c robottelo.properties
```

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]
server.ssh.key_private=[PATH TO YOUR SSH KEY]
server.ssh.username=root
project=foreman
locale=en_US
remote=0
```

(continues on next page)

(continued from previous page)

```
[foreman]
admin.username=admin
admin.password=changeme
```

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.

2.2 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.

2.3 Testing With Nose

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

To run all tests:

```
$ nosetests -c robottelo.properties
```

It is possible to run a specific subset of tests:

```
$ nosetests -c robottelo.properties tests.robottelo.test_decorators
$ nosetests -c robottelo.properties tests.robottelo.test_
↪decorators:DataDecoratorTestCase
$ nosetests -c robottelo.properties tests.robottelo.test_
↪decorators:DataDecoratorTestCase.test_data_decorator_smoke
```

To get more verbose output, or run multiple tests:

```
$ nosetests -c robottelo.properties -v
$ nosetests -c robottelo.properties \
    tests.robottelo.test_decorators \
    tests.robottelo.test_cli
```

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

```
$ nosetests -c robottelo.properties tests.foreman.api
$ nosetests -c robottelo.properties tests.foreman.cli
$ nosetests -c robottelo.properties tests.foreman.ui
```

Many of the existing tests use the DDT module to allow for a more data-driven methodology and in order to run a specific test you need 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 -c robottelo.properties -m test_positive_create_1 \
    tests.foreman.cli.test_org
```

2.4 Running UI Tests On a Headless Server

It is possible to run UI tests on a headless server. To do this:

- Install Xvfb. It is provided by the `xorg-x11-server-Xvfb` package on Fedora and Red Hat.
- Install the `PyVirtualDisplay` Python package. (It is listed in `requirements-optional.txt`.)
- Set `virtual_display=1` in the configuration file `robottelo.properties`.
- Optionally, set the `window_manager_command` option in the configuration file. This option should be set to the name of a window manager executable. For example, `fluxbox` or `openbox` are suitable values if you have Fluxbox or Openbox installed, respectively. Setting this option causes a window manager to be launched before the web browser, thus allowing the web browser to be maximized.

This done, UI tests no longer launch a visible web browser. Instead, UI tests launch a web browser within a virtual display.

CHAPTER 3

API Reference

If you are looking for information on a specific function, class or method, this part of the documentation is for you. The following is an overview of the topics covered by the API. For more granular information, follow one of the links.

3.1 API Documentation

This is the Robottelo API documentation. It is mostly autogenerated from the source code.

3.1.1 robottelo

Submodules:

robottelo.api

Module containing API helper classes. Implementations are in `robottelo/records`.

robottelo.api.apicrud

robottelo.api.base

robottelo.api.client

robottelo.api.utils

robottelo.cli

Submodules:

robottelo.cli.metatest

robottelo.cli.metatest.default_data

robottelo.cli.metatest.template_methods

robottelo.cli.activationkey

robottelo.cli.architecture

robottelo.cli.base

robottelo.cli.computeresource

robottelo.cli.contenthost

robottelo.cli.contentview

robottelo.cli.domain

robottelo.cli.environment

robottelo.cli.factory

robottelo.cli.fact

robottelo.cli.globalparam

robottelo.cli.gpgkey

robottelo.cli.hostcollection

robottelo.cli.hostgroup

robottelo.cli.host

robottelo.cli.lifecycleenvironment

robottelo.cli.medium

robottelo.cli.model

robottelo.cli.operatingsys

robottelo.cli.org

robottelo.cli.partitiontable

robottelo.cli.product

robottelo.cli.proxy

robottelo.cli.puppetmodule

robottelo.cli.puppet

robottelo.cli.report

robottelo.cli.repository

robottelo.cli.smartclass

robottelo.cli.subnet

robottelo.cli.subscription

robottelo.cli.syncplan

robottelo.cli.template

robottelo.cli.user

robottelo.common

Submodules:

robottelo.common.records

robottelo.common.records.base

robottelo.common.records.fields

robottelo.common.constants

robottelo.common.decorators

robottelo.common.helpers

robottelo.common.manifests

robottelo.common.ssh

robottelo.records

robottelo.records.activation_key

robottelo.records.architecture

robottelo.records.content_view_definition

robottelo.records.domain

robottelo.records.environment

robottelo.records.host

robottelo.records.host_collection

robottelo.records.operatingsystem

robottelo.records.organization

robottelo.records.partitiontable

robottelo.records.permission

robottelo.records.product

robottelo.records.repository

robottelo.records.role

robottelo.records.smartproxy

robottelo.records.user

robottelo.ui

robottelo.ui.activationkey

robottelo.ui.architecture

robottelo.ui.base

robottelo.ui.computeresource

robottelo.ui.contentenv

robottelo.ui.contentviews

robottelo.ui.domain

robottelo.ui.environment

robottelo.ui.factory

robottelo.ui.gpgkey

robottelo.ui.hostgroup

robottelo.ui.hosts

robottelo.ui.location

robottelo.ui.locators

robottelo.ui.login

robottelo.ui.medium

robottelo.ui.navigator

robottelo.ui.operatingsys

robottelo.ui.org

robottelo.ui.partitiontable

robottelo.ui.products

robottelo.ui.repository

robottelo.ui.role

robottelo.ui.session

robottelo.ui.subnet

robottelo.ui.subscription

robottelo.ui.syncplan

robottelo.ui.sync

robottelo.ui.systemgroup

robottelo.ui.template

robottelo.ui.usergroup

robottelo.ui.user

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

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

robottelo.entities

robottelo.factory

robottelo.log

robottelo.orm

robottelo.test

robottelo.vm

3.1.2 tests

Submodules:

tests.foreman

Submodules:

tests.foreman.api

tests.foreman.api.test_activationkey

tests.foreman.api.test_activationkey_v2

tests.foreman.api.test_computeresource

tests.foreman.api.test_configtemplate

tests.foreman.api.test_contentviews

tests.foreman.api.test_domain

tests.foreman.api.test_environments

tests.foreman.api.test_foremantask_v2

tests.foreman.api.test_gpgkey_v2

tests.foreman.api.test_hostgroup

tests.foreman.api.test_host

tests.foreman.api.test_host_v2

tests.foreman.api.test_medium

tests.foreman.api.test_multiple_paths

tests.foreman.api.test_org

tests.foreman.api.test_organization_v2

tests.foreman.api.test_permission

tests.foreman.api.test_product

tests.foreman.api.test_repository

tests.foreman.api.test_role_v2

tests.foreman.api.test_smartproxy

tests.foreman.api.test_subnet

tests.foreman.api.test_user

tests.foreman.api.test_user_v2

tests.foreman.cli

tests.foreman.cli.test_activationkey

tests.foreman.cli.test_architecture

tests.foreman.cli.test_computeresource

tests.foreman.cli.test_contenthost

tests.foreman.cli.test_contentviews

tests.foreman.cli.test_domain

tests.foreman.cli.test_environment

tests.foreman.cli.test_fact

tests.foreman.cli.test_globalparam

tests.foreman.cli.test_gpgkey

tests.foreman.cli.test_host_collection

tests.foreman.cli.test_hostgroup

tests.foreman.cli.test_host_system_unification

tests.foreman.cli.test_installer

tests.foreman.cli.test_lifecycleenvironment

tests.foreman.cli.test_medium

tests.foreman.cli.test_model

tests.foreman.cli.test_myaccount

tests.foreman.cli.test_org

tests.foreman.cli.test_os

tests.foreman.cli.test_partitiontable

tests.foreman.cli.test_product

tests.foreman.cli.test_proxy

tests.foreman.cli.test_report

tests.foreman.cli.test_repository

tests.foreman.cli.test_roles

tests.foreman.cli.test_scparam

tests.foreman.cli.test_sso

tests.foreman.cli.test_subnet

tests.foreman.cli.test_subscription

tests.foreman.cli.test_syncplan

tests.foreman.cli.test_template

tests.foreman.cli.test_user

tests.foreman.installer

tests.foreman.installer.test_installer

tests.foreman.smoke

tests.foreman.smoke.test_api_smoke

tests.foreman.smoke.test_cli_smoke

tests.foreman.ui

tests.foreman.ui.test_activationkey

tests.foreman.ui.test_architecture

tests.foreman.ui.test_computeresource

tests.foreman.ui.test_contentenv

tests.foreman.ui.test_contentviews

tests.foreman.ui.test_domain

tests.foreman.ui.test_environment

tests.foreman.ui.test_gpgkey

tests.foreman.ui.test_hostgroup

tests.foreman.ui.test_host

tests.foreman.ui.test_host_system_unification

tests.foreman.ui.test_login

tests.foreman.ui.test_medium

tests.foreman.ui.test_myaccount

tests.foreman.ui.test_operatingsys

tests.foreman.ui.test_org

tests.foreman.ui.test_partitiontable

tests.foreman.ui.test_products

tests.foreman.ui.test_repository

tests.foreman.ui.test_role

tests.foreman.ui.test_sso

tests.foreman.ui.test_subnet

tests.foreman.ui.test_subscription

tests.foreman.ui.test_syncplan

tests.foreman.ui.test_sync

tests.foreman.ui.test_template

tests.foreman.ui.test_usergroup

tests.foreman.ui.test_user

tests.robottelo

Submodules:

tests.robottelo.records

tests.robottelo.records.records

tests.robottelo.records.tests

tests.robottelo.test_factory

tests.robottelo.test_cli

tests.robottelo.test_decorators

tests.robottelo.test_entities

tests.robottelo.test_helpers

tests.robottelo.test_orm

tests.robottelo.test_ssh

tests.robottelo.test_robottelo_api_client

tests.robottelo.test_robottelo_api_inspect

tests.robottelo.test_vm

This module contains tests both for Foreman and Robottelo itself.

Tests for Robottelo are in the `robottelo` module. Those tests are inward-facing: they help ensure that this codebase is functioning correctly.

Tests for Foreman are in the `foreman` module. Those tests are outward-facing: they help ensure that a Foreman deployment is functioning correctly.

CHAPTER 4

Miscellany

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:

```
$ make lint
```

The design and development for this software is led by Og Maciel.

Python Module Index

r

robottelo, 14
robottelo.api, 9
robottelo.cli, 10
robottelo.ui, 12

t

tests, 19
tests.foreman, 18
tests.foreman.api, 14
tests.foreman.cli, 15
tests.foreman.installer, 17
tests.foreman.smoke, 17
tests.foreman.ui, 17
tests.robottelo, 18
tests.robottelo.records, 18

Index

R

`robottelo` (*module*), 14
`robottelo.api` (*module*), 9
`robottelo.cli` (*module*), 10
`robottelo.ui` (*module*), 12

T

`tests` (*module*), 19
`tests.foreman` (*module*), 18
`tests.foreman.api` (*module*), 14
`tests.foreman.cli` (*module*), 15
`tests.foreman.installer` (*module*), 17
`tests.foreman.smoke` (*module*), 17
`tests.foreman.ui` (*module*), 17
`tests.robottelo` (*module*), 18
`tests.robottelo.records` (*module*), 18