

Guidelines for VMware REST module development

The Ansible VMware REST collection (on [Galaxy](#), source code [repository](#)) is maintained by Red Hat and the community.

- [Contribution process](#)
- [Requirements](#)
- [vmware_rest_code_generator](#)
- [Testing with ansible-test](#)
 - [black code formatter](#)
 - [sanity tests](#)
 - [integration tests](#)

Contribution process

The modules of the `vmware_rest` collection are autogenerated by another tool called `vmware_rest_code_generator` <https://github.com/ansible-collections/vmware_rest_code_generator>.

If you would like to contribute a change, we would appreciate if you:

- submit a Github Pull Request (PR) against the `vmware_rest_code_generator` project
- but also ensure the generated modules are compliant with our quality criteria.

Requirements

You will need:

- Python 3.6 or greater
- the `tox` <<https://tox.readthedocs.io/en/latest/install.html>> command

vmware_rest_code_generator

Your contribution should follow the coding style of *Black* <<https://github.com/psf/black>>. To run the code formatter, just run:

```
tox -e black
```

To regenerate the `vmware_rest` collection, you can use the following commands:

```
tox -e refresh_modules -- --target-dir ~/.ansible/collections/ansible_collections/vmware/vmware_rest
```

If you also want to update the EXAMPLE section of the modules, run:

```
tox -e refresh_examples -- --target-dir ~/.ansible/collections/ansible_collections/vmware/vmware_rest
```

Testing with ansible-test

All the modules are covered by a functional test. The tests are located in the `.file:tests/integration/targets/`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\platforms\[ansible-devel] [docs] [docsite] [rst] [dev_guide] [platforms]vmware_rest_guidelines.rst, line 56); [backlink](#)

Unknown interpreted text role "file".

To run the tests, you will need a vcenter instance and an ESXi.

black code formatter

We follow the coding style of *Black* <<https://github.com/psf/black>>. You can run the code formatter with the following command.

```
tox -e black
```

sanity tests

Here we use Python 3.8, the minimal version is 3.6.

```
tox -e black
ansible-test sanity --debug --requirements --local --skip-test future-import-boilerplate --skip-test metaclass-boilerplate
```

integration tests

These tests should be run against your test environment.

..warning:: The test suite will delete all the existing DC from your test environment.

First, prepare a configuration file, we call it `.file:/tmp/inventory-vmware_rest` in this example:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\platforms\[ansible-devel] [docs] [docsite] [rst] [dev_guide] [platforms]vmware_rest_guidelines.rst, line 89); [backlink](#)

Unknown interpreted text role "file".

```
[vmware_rest]
localhost ansible_connection=local ansible_python_interpreter=python

[vmware_rest:vars]
vcenter_hostname=vcenter.test
vcenter_username=administrator@vsphere.local
vcenter_password=kLRy|FXwZSHXW0w?Q:sO
esxi1_hostname=esxi1.test
esxi1_username=zuul
esxi1_password=f6QYNi65k05kv8m56
```

To run the tests, use the following command. You may want to adjust the Python version.

```
ansible-test network-integration --diff --no-temp-workdir --python 3.8 --inventory /tmp/inventory-vmware_rest zuul/
```