
Python vManage

Release 0.3.8

Cisco Public Sector

Oct 14, 2021

CONTENTS:

1	API Reference	3
1.1	utils	3
1.2	data	3
1.3	apps	9
1.4	api	11
2	Indices and tables	51
	Python Module Index	53
	Index	55

python-viptela provides a Python SDK for interacting with Cisco SDWAN (formerly Viptela).

API REFERENCE

This page contains auto-generated API reference documentation¹.

1.1 `utils`

1.1.1 Module Contents

`utils.list_to_dict` (*lst*, *key_name*, *remove_key=True*)

Convert a list of dictionaries into a dictionary of dictionaries.

Parameters

- **key_name** – The name of the key from the child dictionary to use to use in the parent.
- **remove_key** – Remove the key used for the parent dictionary from the child.

Returns The resulting dictionary.

Return type result (dict)

1.2 `data`

1.2.1 Submodules

`data.parse_methods`

Parse Methods for Data Returned by Cisco vManage.

Module Contents

`data.parse_methods.VALID_STATUS_CODES` = [200, 201, 202, 203, 204, 205, 206, 207, 208, 226]

class `data.parse_methods.ParseMethods`

Reset all configuratios on a vManage instance.

Executes the necessary REST calls in specific order to remove configurations applied to a vManage instance.

static `parse_data` (*response*)

Parse data and provide error handling for missing data.

¹ Created with `sphinx-autoapi`

Parameters **response** (*obj*) – Requests response object

Returns All data associated with a response.

Return type result (dict)

Raises **Exception** – Provides error message and details of issue.

static **parse_status** (*response*)

Retrieve status code for transactions that do not receive a response.

Parameters **response** (*obj*) – Requests response object

Returns All data associated with a response.

Return type result (dict)

static **parse_config** (*response*)

Parse config and provide error handling for missing data.

Parameters **response** (*obj*) – Requests response object

Returns All data associated with a response.

Return type result (dict)

Raises **Exception** – Provides error message and details of issue.

static **parse_id** (*response*)

Parse id and provide error handling for missing data.

Parameters **response** (*obj*) – Requests response object

Returns All data associated with a response.

Return type result (dict)

Raises **Exception** – Provides error message and details of issue.

data.policy_data

Cisco vManage Policy Methods.

Module Contents

class data.policy_data.**PolicyData** (*session, host, port=443*)

Bases: object

Methods that deal with importing, exporting, and manipulating data from policies.

import_policy_list_list (*self, policy_list_list, push=False, update=False, check_mode=False, force=False*)

Import a list of policyies lists into vManage. Object Names are translated to IDs.

Parameters

- **policy_list_list** – A list of polcies
- **push** (*bool*) – Whether to push a change out
- **update** (*bool*) – Whether to update when the list exists
- **check_mode** (*bool*) – Report what updates would happen, but don't update

Returns All data associated with a response.

Return type result (dict)

convert_list_name_to_id (*self*, *name_list*)
Convert policy list from names to IDs in object.

Parameters *name_list* (*list*) – Object

convert_list_id_to_name (*self*, *id_list*)
Convert policy list from IDs to names in object.

Parameters *id_list* (*list*) – Object

convert_sequences_to_id (*self*, *sequence_list*)
Convert sequence entries from IDs to names in object.

Parameters *sequence_list* (*list*) – Sequence list

convert_definition_id_to_name (*self*, *policy_definition*)
Convert policy_definition from IDs to names in object.

Parameters *policy_definition* (*list*) – Sequence list

convert_definition_name_to_id (*self*, *policy_definition*)
Convert policy_definition from names to IDs in object.

Parameters *policy_definition* (*list*) – Sequence list

convert_policy_definition_to_name (*self*, *policy_definition*)
Convert policy_definition objects from IDs to names

Parameters *policy_definition* (*list*) – Sequence list

Returns The converted policy definition

Return type result (dict)

convert_policy_definition_to_id (*self*, *policy_definition*)
Convert policy_definition objects from names to IDs

Parameters *policy_definition* (*list*) – Sequence list

Returns The converted policy definition

Return type result (dict)

export_policy_definition_list (*self*, *definition_type*='all')
Export Policy Definition Lists from vManage, translating IDs to Names.

Parameters *definition_type* (*string*) – The type of Definition List to retrieve

Returns

A list of all definition lists currently in vManage.

Return type response (list)

import_policy_definition_list (*self*, *policy_definition_list*, *update*=False, *push*=False, *check_mode*=False, *force*=False)
Import Policy Definitions into vManage. Object names are converted to IDs.

Returns

A list of all policy lists currently in vManage.

Return type response (dict)

convert_policy_to_name (*self*, *policy_item*)
Convert policy items from IDs to names

Parameters `definition_type` (*string*) – Policy item

Returns The converted policy item

Return type response (dict)

convert_policy_to_id (*self*, *policy_item*)

Convert policy items from names IDs

Parameters `definition_type` (*string*) – Policy item

Returns The converted policy item

Return type response (dict)

export_local_policy_list (*self*)

Export Local Policies from vManage. Object IDs are converted to names.

Returns

A list of all policy lists currently in vManage.

Return type response (dict)

import_local_policy_list (*self*, *local_policy_list*, *update=False*, *push=False*,
check_mode=False, *force=False*)

Import Local Policies into vManage. Object names are converted to IDs.

Returns

A list of all policy lists currently in vManage.

Return type response (dict)

export_central_policy_list (*self*)

Export Central Policies from vManage, converting IDs to names.

Returns

A list of all policy lists currently in vManage.

Return type response (dict)

import_central_policy_list (*self*, *central_policy_list*, *update=False*, *push=False*,
check_mode=False, *force=False*)

Import Central Policies into vManage. Object names are converted to IDs.

Returns

A list of all policy lists currently in vManage.

Return type response (dict)

export_security_policy_list (*self*)

Export Security Policies from vManage, converting IDs to names.

Returns

A list of all policy lists currently in vManage.

Return type response (dict)

import_security_policy_list (*self*, *security_policy_list*, *update=False*, *push=False*,
check_mode=False, *force=False*)

Import Security Policies into vManage. Object names are converted to IDs.

Returns

A list of all policy lists currently in vManage.

Return type response (dict)

`data.template_data`

Cisco vManage Templates Methods.

Module Contents

class `data.template_data.TemplateData` (*session, host, port=443*)

Bases: object

Methods that deal with importing, exporting, and converting data from templates.

convert_device_template_to_name (*self, device_template*)

Convert a device template objects from IDs to Names.

Parameters `device_template` (*dict*) – Device Template

Returns Converted Device Template.

Return type result (dict)

convert_device_template_to_id (*self, device_template*)

Convert a device template objects from Names to IDs.

Parameters `device_template` (*dict*) – Device Template

Returns Converted Device Template.

Return type result (dict)

generalTemplates_to_id (*self, generalTemplates*)

Convert a generalTemplates object from Names to IDs.

Parameters `generalTemplates` (*dict*) – generalTemplates object

Returns Converted generalTemplates object.

Return type result (dict)

import_feature_template_list (*self, feature_template_list, push=False, check_mode=False, update=False*)

Import a list of feature templates from list to vManage. Object Names are converted to IDs.

Parameters

- **feature_template_list** (*list*) – List of feature templates
- **check_mode** (*bool*) – Only check to see if changes would be made
- **update** (*bool*) – Update the template if it exists

Returns Returns the diffs of the updates.

Return type result (list)

export_device_template_list (*self, factory_default=False, name_list=None*)

Export device templates from vManage into a list. Object IDs are converted to Names.

Parameters

- **factory_default** (*bool*) – Include factory default
- **name_list** (*list of strings*) – A list of template names to retrieve.

Returns All data associated with a response.

Return type result (dict)

import_device_template_list (*self*, *device_template_list*, *check_mode=False*, *update=False*, *push=False*)

Import a list of device templates from list to vManage. Object Names are converted to IDs.

Parameters

- **device_template_list** (*list*) – List of device templates
- **check_mode** (*bool*) – Only check to see if changes would be made
- **update** (*bool*) – Update the template if it exists

Returns Returns the diffs of the updates.

Return type result (list)

import_attachment_list (*self*, *attachment_list*, *check_mode=False*, *update=False*)

Import a list of device attachments to vManage.

Parameters

- **attachment_list** (*list*) – List of attachments
- **check_mode** (*bool*) – Only check to see if changes would be made
- **update** (*bool*) – Update the template if it exists

Returns Returns the diffs of the updates.

Return type result (list)

subTemplates_to_name (*self*, *old_template*, *feature_template_dict*)

Convert a Sub Template objects from IDs to Names.

Parameters

- **old_template** (*dict*) – a device template
- **feature_template_dict** (*dict*) – dict of all the feature templates

Returns Converted Device Template.

Return type result (dict)

subTemplates_to_id (*self*, *template*, *feature_template_dict*)

Convert a Sub Template objects from IDs to Names.

Parameters

- **template** (*dict*) – a device template
- **feature_template_dict** (*dict*) – dict of all the feature templates

Returns Converted Device Template.

Return type result (dict)

1.2.2 Package Contents

```
data.name = vmanage.data
```

1.3 apps

1.3.1 Submodules

```
apps.clean
```

Clean vManage Resources.

Module Contents

```
class apps.clean.CleanVmanage(session, host, port=443)
    Bases: object

    Reset all configurations on a vManage instance.

    Executes the necessary REST calls in specific order to remove configurations applied to a vManage instance.

    active_count_delay(self)
        Delay while there are active tasks.

    clean_vedge_attachments(self)
        Clean all vedge attachments

    clean_vedge(self)
        Clean all vedges

    clean_controller_attachments(self)
        Clean all controller attachments

    clean_device_templates(self)
        Clean all device templates

    clean_feature_templates(self)
        Clean all feature templates

    clean_central_policy(self)
        Clean all central policy

    clean_local_policy(self)
        Clean all local policy

    clean_policy_definitions(self)
        Clean all policy definitions

    clean_policy_lists(self)
        Clean all policy lists

    clean_security_policy(self)
        Clean all security policy

    clean_all(self)
        Clean everything in vManage
```

apps.files

Cisco vManage Files API Methods.

Module Contents

class `apps.files.Files` (*session, host, port=443*)

Bases: `object`

Read and write data to file.

export_templates_to_file (*self, export_file, name_list=None, template_type=None*)

Export templates to a file. All object IDs will be translated to names. Use a '.yaml' extension to export as YAML and a '.json' extension to export as JSON.

Parameters

- **export_file** (*str*) – The name of the export file
- **name_list** (*list*) – List of device templates to export
- **template_type** (*str*) – Template type: device or template

import_templates_from_file (*self, import_file, update=False, check_mode=False, name_list=None, template_type=None*)

Import templates from a file. All object Names will be translated to IDs.

Parameters

- **import_file** (*str*) – The name of the import file
- **name_list** (*list*) – List of device templates to export
- **template_type** (*str*) – Template type: device or template
- **check_mode** (*bool*) – Try the import, but don't make changes (default: False)
- **update** (*bool*) – Update existing templates (default: False)

export_policy_to_file (*self, export_file*)

Export policy to a file. All object IDs will be translated to names. Use a '.yaml' extension to export as YAML and a '.json' extension to export as JSON.

Parameters **export_file** (*str*) – The name of the export file

import_policy_from_file (*self, file, update=False, check_mode=False, push=False*)

Import policy from a file. All object Names will be translated to IDs.

Parameters

- **import_file** (*str*) – The name of the import file
- **check_mode** (*bool*) – Try the import, but don't make changes (default: False)
- **update** (*bool*) – Update existing templates (default: False)
- **push** (*bool*) – Push templates to devices if changed (default: False)

export_attachments_to_file (*self, export_file, name_list=None, device_type=None*)

Export attachments to a file. All object IDs will be translated to names. Use a '.yaml' extension to export as YAML and a '.json' extension to export as JSON.

Parameters **export_file** (*str*) – The name of the export file

```
import_attachments_from_file (self, import_file, update=False, check_mode=False,  
                                name_list=None)
```

Import attachments from a file. All object Names will be translated to IDs.

Parameters

- **import_file** (*str*) – The name of the import file
- **check_mode** (*bool*) – Try the import, but don't make changes (default: False)
- **update** (*bool*) – Update existing templates (default: False)

1.3.2 Package Contents

```
apps.name = vmanage.apps
```

1.4 api

1.4.1 Submodules

```
api.authentication
```

Cisco vManage Authentication API Methods.

Module Contents

```
class api.authentication.Authentication (host=None, user=None, password=None,  
                                           port=443, validate_certs=False, timeout=10)
```

Bases: object

vManage Authentication API

Responsible for retrieving the JSESSIONID after a username/password has been authenticated. If the vManage version is >= 19.2.0 then the X-XSRF-TOKEN will be retrieved and added to the header. An HTTP(S) Request session object will be returned.

```
login (self)
```

Executes login tasks against vManage to retrieve token(s).

Parameters None. –

Returns a Requests session with JSESSIONID and an X-XSRF-TOKEN for vManage version >= 19.2.0.

Return type self.session

Raises

- **LoginFailure** – If the username/password are incorrect.
- **RequestException** – If the host is not accessible.

api.central_policy

Cisco vManage Centralized Policy API Methods.

Module Contents

class api.central_policy.CentralPolicy (session, host, port=443)

Bases: object

vManage Central Policy API

Responsible for DELETE, GET, POST, PUT methods against vManage Central Policy.

activate_central_policy (self, policy_name, policy_id)

Activates the current active centralized policy

Parameters policyId (str) – ID of the active centralized policy

Returns The Action ID from the activation.

Return type action_id (str)

reactivate_central_policy (self, policy_id)

reActivates the current active centralized policy

Parameters policyId (str) – ID of the active centralized policy

Returns The Action ID from the activation.

Return type action_id (str)

deactivate_central_policy (self, policy_id)

Deactivates the current active centralized policy

Parameters policyId (str) – ID of the deactive centralized policy

Returns The Action ID from the activation.

Return type result (str)

add_central_policy (self, policy)

Delete a Central Policy from vManage.

Parameters policy – The Central Policy

Returns All data associated with a response.

Return type result (dict)

update_central_policy (self, policy, policy_id)

Update a Central from vManage.

Parameters

- **policy** – The Central Policy

- **policy_id** – The ID of the Central Policy to update

Returns All data associated with a response.

Return type result (dict)

delete_central_policy (self, policyId)

Deletes the specified centralized policy

Parameters policyId (str) – ID of the active centralized policy

Returns All data associated with a response.

Return type result (dict)

get_central_policy (*self*)

Obtain a list of all configured central policies

Returns All data associated with a response.

Return type result (dict)

get_central_policy_list (*self*)

Get all Central Policies from vManage.

Returns

A list of all policy lists currently in vManage.

Return type response (list)

get_central_policy_dict (*self*, *key_name*='policyName', *remove_key*=False)

Get all Central Policies from vManage.

Parameters

- **key_name** (*str*) – The name of the attribute to use as the key
- **remove_key** (*bool*) – Remove the key from the dict (default: False)

Returns

A dict of all Central Policies currently in vManage.

Return type response (dict)

api.certificate

Cisco vManage Certificate API Methods.

Module Contents

class api.certificate.Certificate (*session*, *host*, *port*=443)

Bases: object

vManage Certificate API

Responsible for DELETE, GET, POST, PUT methods against vManage Certificates.

generate_csr (*self*, *device_ip*)

Generate CSR for device

Parameters **device_ip** (*str*) – IP address of device.

Returns The CSR for the device.

Return type deviceCSR (str)

install_device_cert (*self*, *cert*)

Install signed cert on vManage

Parameters **cert** (*str*) – The certificate to install.

Returns The action ID of the install command.

Return type id (str)

push_certificates (*self*)

Push certificates to all controllers

Returns The action ID of the push command.

Return type id (str)

get_vmanage_root_cert (*self*)

Get vManage root certificate

Args:

Returns The root certificate.

Return type rootcertificate (str)

api.cluster

Cisco vManage Cluster API Methods.

Module Contents

class `api.cluster.Cluster` (*session, host, port=443*)

Bases: object

vManage Cluster API

Responsible for DELETE, GET, POST, PUT methods against vManage Cluster.

get_cluster_connected_devices_list (*self, vmanage_cluster_ip*)

Obtain vManage cluster connected devices

Parameters `vmanage_cluster_ip` (*str*) – vManage cluster interface IP address

Returns All data associated with a response.

Return type result (list)

get_cluster_health_details_list (*self*)

Obtain vManage cluster health details

Parameters `None` (*None*) –

Returns All data associated with a response.

Return type result (list)

get_cluster_health_status_list (*self*)

Obtain vManage cluster health status

Parameters `None` (*None*) –

Returns All data associated with a response.

Return type result (list)

get_cluster_list (*self*)

Obtain vManage cluster list

Parameters `None` (*None*) –

Returns All data associated with a response.

Return type result (list)

get_cluster_ip_addresses_dict (*self*)

Obtain vManage cluster IP addresses

Parameters **None** (*None*) –

Returns All data associated with a response.

Return type result (dict)

get_cluster_ready_state (*self*)

Obtain vManage cluster ready state

Parameters **None** (*None*) –

Returns All data associated with a response.

Return type result (bool)

get_cluster_node_properties (*self*)

Obtain connected vManage cluster node properties

Parameters **None** (*None*) –

Returns All data associated with a response.

Return type result (dict)

get_cluster_tenancy_mode (*self*)

Obtain vManage cluster tenancy mode

Parameters **None** (*None*) –

Returns All data associated with a response.

Return type result (dict)

get_cluster_vmanage_details_list (*self*, *vmanage_cluster_ip*)

Obtain vManage cluster specific vManage details using cluster interface IP

Parameters **vmanage_cluster_ip** (*str*) – vManage cluster interface IP address

Returns All data associated with a response.

Return type result (list)

api.device

Cisco vManage Device Inventory API Methods.

Module Contents

class `api.device.Device` (*session*, *host*, *port=443*)

Bases: `object`

vManage Device Inventory API

Responsible for DELETE, GET, POST, PUT methods against vManage Device Inventory.

get_device_list (*self*, *category*)

Obtain a list of specified device type

Parameters **category** (*str*) – vedges or controllers

Returns All data associated with a response.

Return type result (dict)

post_device_cli_mode (*self*, *deviceId*, *deviceType*)

Update a device to CLI mode

Parameters

- **deviceId** (*str*) – uuid for device object
- **deviceType** (*str*) – vedge or controller

get_device_status_list (*self*)

Obtain a list of specified device type

Args: None

Returns Device status

Return type result (list)

get_device_status_dict (*self*, *key_name*='host-name', *remove_key*=False)

Obtain a dict of specified device type

Parameters **category** (*str*) – vedges or controllers

Returns Device status

Return type result (dict)

get_device_status (*self*, *value*, *key*='system-ip')

Get the status of a specific device

Parameters

- **string** (*value*) – The value of the key to match
- **key** (*string*) – The key on which to match (e.g. system-ip)

Returns Device status

Return type result (dict)

get_device_config (*self*, *device_type*, *value*, *key*='system-ip')

Get the config of a specific device

Parameters

- **string** (*value*) – The value of the key to match
- **key** (*string*) – The key on which to match (e.g. system-ip)

Returns Device config

Return type result (dict)

get_device_config_list (*self*, *device_type*)

Get the config status of a list of devices. When ‘all’ is specified, it concatenats the vedges and controller together to provide a single method to retrieve status in the same way as `get_device_status_list`.

Parameters **device_type** (*str*) – ‘vedges’, ‘controllers’, or ‘all’

Returns All data associated with a response.

Return type result (list)

get_device_config_dict (*self*, *device_type*, *key_name*='host-name', *remove_key*=False)

Get the config status of a list of devices as a dictionary

Parameters

- **device_type** (*str*) – vedge or controller
- **key_name** (*string*) – The name of the attribute to use as the dictionary key
- **remove_key** (*boolean*) – remove the search key from the element

Returns All data associated with a response.

Return type result (dict)

get_device_data (*self, path, device_ip*)

Get the data from a device

Parameters

- **path** (*str*) – The path of the data
- **device_ip** (*str*) – The IP address of the device

Returns All data associated with a response.

Return type result (dict)

put_device_decommission (*self, device_id*)

Decommission a device

Parameters **device_id** (*str*) – uuid for device object

Returns Device status

Return type result (list)

post_device (*self, device_ip, personality, username, password*)

Add control plane device

Parameters

- **device_ip** (*str*) – device interface IP
- **personality** (*str*) – controller type (vmanage, vsmart, vbond)
- **username** (*str*) – device username
- **password** (*str*) – device password

Returns Device status

Return type result (list)

post_reset_interface (*self, device_ip, vpn_id, ifname*)

Reset an Interface :param device_ip: device IP for device object :type device_ip: str :param vpn_id: VPN Id for Interface :type vpn_id: int :param ifname: Interface name to reset :type ifname: str

Returns HTTP response status

Return type result (int)

api.device_templates

Cisco vManage Device Templates API Methods.

Module Contents

```
class api.device_templates.DeviceTemplates (session, host, port=443)
    Bases: object
    vManage Device Templates API
    Responsible for DELETE, GET, POST, PUT methods against vManage Device Templates.

delete_device_template (self, templateId)
    Obtain a list of all configured device templates.

        Parameters templateId (str) – Object ID for device template

        Returns All data associated with a response.

        Return type result (dict)

get_device_templates (self)
    Obtain a list of all configured device templates.

        Returns All data associated with a response.

        Return type result (dict)

get_device_template_object (self, template_id)
    Obtain a device template object.

        Parameters template_id (str) – Object ID for device template

        Returns All data associated with a response.

        Return type result (dict)

get_device_template_list (self, factory_default=False, name_list=None)
    Get the list of device templates.

        Parameters

            • factory_default (bool) – Include factory default

            • name_list (list of strings) – A list of template names to retrieve.

        Returns All data associated with a response.

        Return type result (dict)

get_device_template_dict (self, factory_default=False, key_name='templateName', remove_key=True, name_list=None)
    Obtain a dictionary of all configured device templates.

        Parameters

            • factory_default (bool) – Wheter to return factory default templates

            • key_name (string) – The name of the attribute to use as the dictionary key

            • remove_key (boolean) – remove the search key from the element

            • name_list (list of strings) – A list of template names to retrieve.

        Returns All data associated with a response.
```

Return type result (dict)

get_template_attachments (*self*, *template_id*, *key*='host-name')

Get the devices that a template is attached to.

Parameters

- **template_id** (*string*) – Template ID
- **key** (*string*) – The key of the device to put in the list (default: host-name)

Returns List of keys.

Return type result (list)

get_template_input (*self*, *template_id*, *device_id_list*=None)

Get the input associated with a device attachment.

Parameters

- **template_id** (*string*) – Template ID
- **device_id_list** (*list*) – list of device UUID's to get input for

Returns All data associated with a response.

Return type result (dict)

add_device_template (*self*, *device_template*)

Add a single device template to Vmanage.

Parameters **device_template** (*dict*) – Device Template

Returns Response from Vmanage

Return type result (list)

update_device_template (*self*, *device_template*)

Update a single device template to Vmanage.

Parameters **device_template** (*dict*) – Device Template

Returns Response from Vmanage

Return type result (list)

reattach_device_template (*self*, *template_id*, *config_type*, *is_edited*=True, *is_master_edited*=True)

Re-Attach a template to the devices it it attached to.

Parameters

- **template_id** (*str*) – The template ID to attach to
- **config_type** (*str*) – Type of template i.e. device or CLI template
- **is_edited** (*bool*) – True if the template has been edited
- **is_master_edited** (*bool*) – For CLI device template needs to match is_edited. For device templates using feature templates needs to be set to False.

Returns Returns the action id of the attachment

Return type action_id (str)

attach_to_template (*self*, *template_id*, *config_type*, *uuid*)

Attach and device to a template

Parameters

- **template_id** (*str*) – The template ID to attach to
- **config_type** (*str*) – Type of template i.e. device or CLI template
- **uuid** (*dict*) – The UUIDs of the device to attach and mapping for corresponding variables, system-ip, host-name

Returns Returns the action id of the attachment

Return type action_id (str)

detach_from_template (*self, uuid, device_ip, device_type*)

Detach a device from a template (i.e. Put in CLI mode)

Parameters

- **uuid** (*str*) – The UUID of the device to detach
- **device_ip** (*str*) – The System IP of the system to detach
- **device_type** (*str*) – The device type of the device to detach

Returns Returns the action id of the attachment

Return type action_id (str)

get_attachments (*self, template_id, key='host-name'*)

Get a list of attachments to a particular template.

Parameters

- **template_id** (*str*) – Template ID of the template
- **key** (*str*) – The key of the elements to return

Returns Returns the specified key of the attached devices.

Return type result (list)

get_device_running_config (*self, uuid*)

Get the running configuration of a specific device.

Parameters **uuid** (*str*) – UUID of device

Returns The running configuration of the specified device.

Return type result (str)

reattach_multi_device_templates (*self, template_ids*)

Re-Attach a template to the devices it it attached to.

Parameters

- **template_id** (*str*) – The template ID to attach to
- **config_type** (*str*) – Type of template i.e. device or CLI template
- **is_edited** (*bool*) – True if the template has been edited
- **is_master_edited** (*bool*) – For CLI device template needs to match is_edited. For device templates using feature templates needs to be set to False.

Returns Returns the action id of the attachment

Return type action_id (str)

get_multi_attach_payload (*self, template_ids*)

api.feature_templates

Cisco vManage Feature Templates API Methods.

Module Contents

class `api.feature_templates.FeatureTemplates` (*session, host, port=443*)

Bases: `object`

vManage Feature Templates API

Responsible for DELETE, GET, POST, PUT methods against vManage Feature Templates.

delete_feature_template (*self, templateId*)

Obtain a list of all configured feature templates.

Parameters `templateId` (*str*) – Object ID for feature template

Returns All data associated with a response.

Return type `result` (`dict`)

get_feature_templates (*self*)

Obtain a list of all configured feature templates.

Returns All data associated with a response.

Return type `result` (`dict`)

add_feature_template (*self, feature_template*)

Add a feature template to Vmanage.

Parameters `feature_template` (*dict*) – Feature Template

Returns Response from Vmanage

Return type `result` (`list`)

update_feature_template (*self, feature_template*)

Update a feature template on Vmanage.

Parameters `feature_template` (*dict*) – Feature Template

Returns Response from Vmanage

Return type `result` (`list`)

get_feature_template_list (*self, factory_default=False, name_list=None*)

Obtain a list of all configured feature templates.

Parameters

- **factory_default** (*bool*) – Whether to return factory default templates
- **name_list** (*list of strings*) – A list of the template names to return

Returns All data associated with a response.

Return type `result` (`dict`)

get_feature_template_dict (*self, factory_default=False, key_name='templateName', remove_key=True, name_list=None*)

Obtain a dictionary of all configured feature templates.

Parameters

- **factory_default** (*bool*) – Whether to return factory default templates
- **key_name** (*string*) – The name of the attribute to use as the dictionary key
- **remove_key** (*boolean*) – Remove the search key from the element
- **name_list** (*list of strings*) – A list of the template names to return

Returns All data associated with a response.

Return type result (dict)

get_device_templates_for_feature (*self, templateId*)

Obtain a list of device templates for given feature template

Parameters **templateId** – (str) Feature template ID to find device templates for

Returns All data associated with a response.

Return type result (dict)

`api.http_methods`

Module Contents

`api.http_methods.STANDARD_HEADERS`

`api.http_methods.STANDARD_TIMEOUT = 10`

`api.http_methods.VALID_STATUS_CODES = [200, 201, 202, 203, 204, 205, 206, 207, 208, 226]`

class `api.http_methods.HttpMethods` (*session, url*)

Bases: object

HTTP Methods for vManage API Interaction

Provides a consistent interaction with the vManage REST API. Contains error handling for common HTTP interaction issues.

request (*self, method, headers=None, payload=None, files=None, timeout=STANDARD_TIMEOUT*)

Performs HTTP REST API Call.

Parameters

- **method** (*str*) – DELETE, GET, POST, PUT
- **headers** (*dict*) – Use standard vManage header provided in module or custom header for specific API interaction
- **payload** (*str*) – A formatted string to be delivered to vManage via POST or PUT REST call
- **file** (*obj*) – A file to be sent to vManage

Returns

A parsable dictionary containing the full response from vManage for an interaction

Return type result (dict)

Raises

- **JSONDecodeError** – Payload format error.
- **ConnectionError** – Connection error.

- **HTTPError** – An HTTP error occurred.
- **URLRequired** – A valid URL is required to make a request.
- **TooManyRedirects** – Too many redirects.
- **Timeout** – The request timed out.
- **RequestException** – There was an ambiguous exception.

`api.local_policy`

Cisco vManage Localized Policy API Methods.

Module Contents

class `api.local_policy.LocalPolicy` (*session, host, port=443*)

Bases: `object`

vManage Local Policy API

Responsible for DELETE, GET, POST, PUT methods against vManage Local Policy.

add_local_policy (*self, policy*)

Delete a Central Policy from vManage.

Parameters *policy* – The Central Policy

Returns All data associated with a response.

Return type `result` (`dict`)

update_local_policy (*self, policy, policy_id*)

Update a Central from vManage.

Parameters

- *policy* – The Central Policy
- *policy_id* – The ID of the Central Policy to update

Returns All data associated with a response.

Return type `result` (`dict`)

delete_local_policy (*self, policy_id*)

Deletes the specified local policy

Parameters *policyId* (*str*) – ID of the active local policy

Returns All data associated with a response.

Return type `result` (`dict`)

get_local_policy (*self*)

Obtain a list of all configured local policies

Returns All data associated with a response.

Return type `result` (`dict`)

get_local_policy_list (*self*)

Get all Central Policies from vManage.

Returns

A list of all policy lists currently in vManage.

Return type response (dict)

get_local_policy_dict (*self*, *key_name*='policyName', *remove_key*=False)

Get all Local Policies from vManage.

Parameters

- **key_name** (*str*) – The name of the attribute to use as the key
- **remove_key** (*bool*) – Remove the key from the dict (default: False)

Returns

A dict of all Local Policies currently in vManage.

Return type response (dict)

`api.monitor_network`

Cisco vManage Monitor Networks API Methods.

Module Contents

class `api.monitor_network.MonitorNetwork` (*session*, *host*, *port*=443)

Bases: object

vManage Monitor Networks API

Responsible for GET methods against vManage Real Time Monitoring for network devices.

_get_device_type (*self*, *system_ip*)

_ipsec_query_params (*self*, *query_params*, ***kwargs*)

get_aaa_users (*self*, *system_ip*)

Provides AAA users for device.

Parameters **system_ip** (*str*) – Device System IP

Returns Device ARP table data.

Return type result (dict)

get_arp_table (*self*, *system_ip*)

Provides ARP entries for device.

Parameters **system_ip** (*str*) – Device System IP

Returns Device ARP table data.

Return type result (dict)

get_bfd_history (*self*, *system_ip*, ***kwargs*)

Provides BFD history for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **remote_system_ip** (*str*) – (Optional) Remote System IP
- **remote_color** (*str*) – (Optional) Remote Color

Returns Device BFD history data.

Return type result (dict)

get_bfd_links (*self*, *system_ip*, ***kwargs*)
Provides BFD links for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **state** (*str*) – (Optional) State

Returns Device BFD links data.

Return type result (dict)

get_bfd_sessions (*self*, *system_ip*, ***kwargs*)
Provides BFD sessions for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **remote_system_ip** (*str*) – (Optional) Remote System IP
- **remote_color** (*str*) – (Optional) Remote Color
- **local_color** (*str*) – (Optional) Local Color

Returns Device BFD sessions data.

Return type result (dict)

get_bfd_device_state (*self*, *system_ip*)
Provides BFD state for device.

Parameters **system_ip** (*str*) – Device System IP

Returns Device BFD state data.

Return type result (dict)

get_bfd_device_state_tloc (*self*, *system_ip*)
Provides BFD state summary with TLOC for device.

Parameters **system_ip** (*str*) – Device System IP

Returns Device BFD state summary with TLOC data.

Return type result (dict)

get_bfd_summary (*self*, *system_ip*)
Provides BFD summary data for device.

Parameters **system_ip** (*str*) – Device System IP

Returns Device BFD summary data.

Return type result (dict)

get_bfd_tloc (*self*, *system_ip*)
Provides BFD TLOC for device.

Parameters **system_ip** (*str*) – Device System IP

Returns Device BFD TLOC data.

Return type result (dict)

get_bgp_neighbors (*self*, *system_ip*, ***kwargs*)
Provides BGP neighbors for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **vpn-id** (*str*) – VPN ID
- **peer-addr** (*str*) – Peer address
- **as** (*str*) – ASN

Returns All data associated with a response.

Return type result (dict)

get_bgp_routes (*self*, *system_ip*, ***kwargs*)
Provides BGP routes for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **vpn-id** (*str*) – VPN ID
- **prefix** (*str*) – IP prefix
- **nexthop** (*str*) – Nexthop

Returns All data associated with a response.

Return type result (dict)

get_bgp_summary (*self*, *system_ip*)
Provides BGP summary for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_cellular_connections (*self*, *system_ip*)
Provides cellular connection information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_cellular_hardware (*self*, *system_ip*)
Provides cellular hardware information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_cellular_modems (*self*, *policy_id*)
Provides cellular modem information for device.

Parameters **policy_id** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_cellular_networks (*self, system_ip*)

Provides cellular network information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_cellular_profiles (*self, system_ip*)

Provides cellular profile information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_cellular_radios (*self, system_ip*)

Provides cellular radio information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_cellular_sessions (*self, system_ip, **kwargs*)

Provides cellular session information for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **interface_name** (*str*) – Interface name values: ge0/0 - ge0/7, system, , eth0
- **primary_dns_ipv4** (*str*) – Primary DNS IP

Returns All data associated with a response.

Return type result (dict)

get_cellular_status (*self, system_ip*)

Provides cellular status information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_control_affinity_config (*self, system_ip*)

Provides current control affinity config for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_control_affinity_status (*self, system_ip*)

Provides current control affinity status for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_control_connections (*self*, *system_ip*, ***kwargs*)

Provides current control connections for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **peer_type** (*str*) – Peer type (vedge, vsmart, vmanage, vbond)
- **peer_system_ip** (*str*) – Peer System IP

Returns All data associated with a response.

Return type result (dict)

get_control_connections_history (*self*, *system_ip*, ***kwargs*)

Provides control connections history for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **peer_type** (*str*) – Peer type (vedge, vsmart, vmanage, vbond)
- **peer_system_ip** (*str*) – Peer System IP
- **local_color** (*str*) – Local Color

Returns All data associated with a response.

Return type result (dict)

get_control_count (*self*)

Provides current control count for device.

Parameters **None** (*None*) –

Returns All data associated with a response.

Return type result (dict)

get_control_links (*self*, *system_ip*, ***kwargs*)

Provides control links for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **state** (*str*) – State

Returns All data associated with a response.

Return type result (dict)

get_control_local_properties (*self*, *system_ip*)

Provides control local properties history for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_control_summary (*self*, *system_ip*)

Provides current control summary for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_control_valid_devices (*self*, *system_ip*)

Provides control valid devices for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_control_valid_vsmarts (*self*, *system_ip*)

Provides control valid vsmarts for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_control_wan_interface (*self*, *system_ip*)

Provides current control wan interface for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_control_wan_interface_color (*self*, *system_ip*)

Provides current control wan interface for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_device_status (*self*, *system_ip*)

Provides status for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_device_system_info (*self*, *system_ip*)

Provides system info for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_dhcp_clients (*self*, *system_ip*)

Provides dhcp clients for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_dhcp_interfaces (*self*, *system_ip*)

Provides dhcp interfaces for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_dhcp_servers (*self*, *system_ip*)

Provides dhcp servers for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_dot1x_clients (*self*, *system_ip*)

Provides DOT1X clients for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_dot1x_interfaces (*self*, *system_ip*)

Provides DOT1X interfaces for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_dot1x_radius (*self*, *system_ip*)

Provides DOT1X RADIUS information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_dpi_applications (*self*, *system_ip*, ***kwargs*)

Provides DPI application information for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **vpn_id** (*str*) – VPN/VRF ID
- **application** (*str*) – Application
- **family** (*str*) – Family

Returns All data associated with a response.

Return type result (dict)

get_dpi_common_applications (*self*)

Provides common DPI application information.

Parameters **None** (*None*) – None

Returns All data associated with a response.

Return type result (dict)

get_dpi_device_fields (*self*)

Provides DPI fields for device.

Parameters **None** (*None*) – None

Returns All data associated with a response.

Return type result (dict)

get_dpi_device_fields_details (*self*)

Provides DPI detailed fields for device.

Parameters **None** (*None*) – None

Returns All data associated with a response.

Return type result (dict)

get_dpi_flows (*self*, *system_ip*, ***kwargs*)

Provides DPI application information for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **vpn_id** (*str*) – VPN/VRF ID
- **source_ip** (*str*) – Source IP
- **application** (*str*) – Application
- **family** (*str*) – Family

Returns All data associated with a response.

Return type result (dict)

get_dpi_summary (*self*, *system_ip*)

Provides DPI summary information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_dpi_supported_applications (*self*, *system_ip*, ***kwargs*)

Provides DPI supported application information for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **application** (*str*) – Application
- **family** (*str*) – Family

Returns All data associated with a response.

Return type result (dict)

get_eigrp_interfaces (*self*, *system_ip*)

Provides EIGRP interface for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_eigrp_routes (*self*, *system_ip*)

Provides EIGRP route for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_eigrp_topology (*self*, *system_ip*)

Provides EIGRP topology for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_hardware_alarms (*self*, *system_ip*)

Provides hardware alarms for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_hardware_environment (*self*, *system_ip*)

Provides hardware environment information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_hardware_inventory (*self*, *system_ip*)

Provides hardware inventory information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_hardware_status_summary (*self*, *system_ip*)

Provides hardware status summary information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_hardware_system (*self*, *system_ip*)

Provides hardware system information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_igmp_groups (*self*, *system_ip*)

Provides IGMP groups information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_igmp_interfaces (*self*, *system_ip*)

Provides IGMP interfaces information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_igmp_statistics (*self*, *system_ip*)

Provides IGMP statistics information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_igmp_summary (*self*, *system_ip*)

Provides IGMP summary information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_interface_vpn (*self*, *system_ip*)

Provides interface vpn/vrf information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_ip_fib (*self*, *system_ip*, ***kwargs*)

Provides IP FIB for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **vpn_id** (*str*) – VPN ID
- **address_family** (*str*) – Address Family
- **destination_prefix** (*str*) – Destination Prefix
- **tloc** (*str*) – TLOC
- **color** (*str*) – TLOC color

Returns All data associated with a response.

Return type result (dict)

get_ip_route_table (*self*, *system_ip*, ***kwargs*)

Provides IP route table for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **vpn_id** (*str*) – VPN ID
- **address_family** (*str*) – Address Family
- **destination_prefix** (*str*) – Destination Prefix

- **source_protocol** (*str*) – Source Protocol
- **next_hop_address** (*str*) – Next-Hop Address (cEdge Only)
- **next_hop_oif** (*str*) – Next-Hop Outgoing Interface (cEdge Only)

Returns All data associated with a response.

Return type result (dict)

get_ip_nat_translations (*self*, *system_ip*)

Provides IP NAT translations for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_ip_nat64_translations (*self*, *system_ip*)

Provides IP NAT64 translations for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_ipsec_inbound_connections (*self*, *system_ip*, ***kwargs*)

Provides IPSec inbound connections for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **remote_tloc_address** (*str*) – Remote TLOC address
- **remote_tloc_color** (*str*) – Remote TLOC color
- **local_tloc_color** (*str*) – Local TLOC color

Returns All data associated with a response.

Return type result (dict)

get_ipsec_localsa (*self*, *system_ip*)

Provides IPSec localsa for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_ipsec_outbound_connections (*self*, *system_ip*, ***kwargs*)

Provides IPSec outbound connections for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **remote_tloc_address** (*str*) – Remote TLOC address
- **remote_tloc_color** (*str*) – Remote TLOC color

Returns All data associated with a response.

Return type result (dict)

get_ipsec_pwk_inbound_connections (*self*, *system_ip*, ***kwargs*)
Provides IPSec pairwise key inbound connections for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **remote_tloc_address** (*str*) – Remote TLOC address
- **remote_tloc_color** (*str*) – Remote TLOC color
- **local_tloc_color** (*str*) – Local TLOC color

Returns All data associated with a response.

Return type result (dict)

get_ipsec_pwk_localsa (*self*, *system_ip*, ***kwargs*)
Provides IPSec pairwise key localsa for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **remote_tloc_address** (*str*) – Remote TLOC address
- **remote_tloc_color** (*str*) – Remote TLOC color
- **local_tloc_color** (*str*) – Local TLOC color

Returns All data associated with a response.

Return type result (dict)

get_ipsec_pwk_outbound_connections (*self*, *system_ip*, ***kwargs*)
Provides IPSec pairwise key outbound connections for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **remote_tloc_address** (*str*) – Remote TLOC address
- **remote_tloc_color** (*str*) – Remote TLOC color
- **local_tloc_color** (*str*) – Local TLOC color

Returns All data associated with a response.

Return type result (dict)

get_multicast_replicator (*self*, *system_ip*)
Provides multicast replicator for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_multicast_rpf (*self*, *system_ip*)
Provides multicast rpf for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_multicast_topology (*self*, *system_ip*)

Provides multicast topology for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_multicast_tunnel (*self*, *system_ip*)

Provides multicast tunnel for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_omp_peers (*self*, *system_ip*)

Provides OMP peers for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_omp_routes_received (*self*, *system_ip*, ****kwargs**)

Provides OMP received routes for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **vpn_id** (*str*) – VPN ID
- **prefix** (*str*) – Prefix

Returns All data associated with a response.

Return type result (dict)

get_omp_routes_advertised (*self*, *system_ip*, ****kwargs**)

Provides OMP advertised routes for device.

Parameters

- **system_ip** (*str*) – Device System IP
- **vpn_id** (*str*) – VPN ID
- **prefix** (*str*) – Prefix

Returns All data associated with a response.

Return type result (dict)

get_omp_summary (*self*, *system_ip*)

Provides OMP summary for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_orchestrator_summary (*self*, *system_ip*)

Provides orchestrator (vbond) summary for device.

Parameters `system_ip` (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_orchestrator_connections (*self*, *system_ip*)

Provides orchestrator (vbond) connections for device.

Parameters `system_ip` (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_orchestrator_connections_history (*self*, *system_ip*)

Provides orchestrator (vbond) connections history for device.

Parameters `system_ip` (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_orchestrator_local_properties (*self*, *system_ip*)

Provides orchestrator (vbond) local properties history for device.

Parameters `system_ip` (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_ospf_interfaces (*self*, *system_ip*)

Provides OSPF interfaces for device.

Parameters `system_ip` (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_ospf_neighbors (*self*, *system_ip*)

Provides OSPF neighbors for device.

Parameters `system_ip` (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_ospf_routes (*self*, *system_ip*)

Provides OSPF routes for device.

Parameters `system_ip` (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_ospf_database (*self*, *system_ip*)

Provides OSPF database for device.

Parameters `system_ip` (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_ospf_database_summary (*self, system_ip*)

Provides OSPF database summary for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_ospf_process (*self, system_ip*)

Provides OSPF process for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_ospf_database_external (*self, system_ip*)

Provides OSPF database external for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_pim_interfaces (*self, system_ip*)

Provides PIM interfaces for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_pim_neighbors (*self, system_ip*)

Provides PIM neighbors for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_pim_rp_mapping (*self, system_ip*)

Provides PIM RP mapping for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_pim_statistics (*self, system_ip*)

Provides PIM statistics for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_security_information (*self, system_ip*)

Provides security information for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_software (*self*, *system_ip*)

Provides software for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_umbrella_sig_tunnels (*self*, *system_ip*)

Provides Cisco Umbrella SIG tunnels for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_users (*self*, *system_ip*)

Provides users for device.

Parameters **system_ip** (*str*) – Device System IP

Returns Device ARP table data.

Return type result (dict)

get_users_list (*self*, *system_ip*)

Provides users list for device.

Parameters **system_ip** (*str*) – Device System IP

Returns Device ARP table data.

Return type result (dict)

get_vrrp (*self*, *system_ip*)

Provides VRRP entries for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_wlan_clients (*self*, *system_ip*)

Provides WLAN clients for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_wlan_interfaces (*self*, *system_ip*)

Provides WLAN interfaces for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_wlan_radios (*self*, *system_ip*)

Provides WLAN radios for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_wlan_radius (*self*, *system_ip*)

Provides WLAN RADIUS authentication for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

get_zscaler_sig_tunnels (*self*, *system_ip*)

Provides Cisco zScaler SIG tunnels for device.

Parameters **system_ip** (*str*) – Device System IP

Returns All data associated with a response.

Return type result (dict)

api.policy_definitions

Cisco vManage Policy Definitions API Methods.

Module Contents

api.policy_definitions.definition_types_19_2_0 = ['data', 'approute', 'control', 'cflowd',

api.policy_definitions.definition_types_19_3_0 = ['data', 'approute', 'control', 'cflowd',

api.policy_definitions.all_definition_types

class api.policy_definitions.PolicyDefinitions (*session*, *host*, *port=443*)

Bases: object

vManage Policy Definitions API

Responsible for DELETE, GET, POST, PUT methods against vManage Policy Definitions used in Centralized, Localized, and Security Policy.

get_definition_types (*self*)

Return the definition types for this version of vManage

Args:

Returns List of definition types

Return type result (list)

delete_policy_definition (*self*, *definition_type*, *definition_id*)

Delete a Policy Definition from vManage.

Parameters

- **definition_type** (*str*) – The defintion type of the requested policy definition
- **definition_id** (*str*) – The defintion ID of the requested policy definition

Returns All data associated with a response.

Return type result (dict)

add_policy_definition (*self, policy_definition*)

Delete a Policy Definition from vManage.

Parameters

- **definition_type** (*str*) – The definition type of the requested policy definition
- **definition_id** (*str*) – The definition ID of the requested policy definition

Returns All data associated with a response.

Return type result (dict)

update_policy_definition (*self, policy_definition, policy_definition_id*)

Update a Policy Definition from vManage.

Parameters

- **definition_type** (*str*) – The definition type of the requested policy definition
- **definition_id** (*str*) – The definition ID of the requested policy definition

Returns All data associated with a response.

Return type result (dict)

get_policy_definition (*self, definition_type, definition_id*)

Get a Policy Definition from vManage.

Parameters

- **definition_type** (*str*) – The definition type of the requested policy definition
- **definition_id** (*str*) – The definition ID of the requested policy definition

Returns All data associated with a response.

Return type result (dict)

get_policy_definition_list (*self, definition_type='all'*)

Get all Policy Definition Lists from vManage.

Parameters **definition_type** (*string*) – The type of Definition List to retrieve

Returns

A list of all definition lists currently in vManage.

Return type response (dict)

get_policy_definition_dict (*self, definition_type, key_name='name', remove_key=False*)

Get all Policy Definition Lists from vManage.

Parameters

- **definition_type** (*str*) – Policy definition type
- **key_name** (*string*) – The name of the attribute to use as the dictionary key
- **remove_key** (*boolean*) – Remove the search key from the element

Returns All data associated with a response.

Return type result (dict)

api.policy_lists

Cisco vManage Policy Lists API Methods.

Module Contents

class `api.policy_lists.PolicyLists` (*session, host, port=443*)

Bases: `object`

vManage Policy Lists API

Responsible for DELETE, GET, POST, PUT methods against vManage Policy Lists used in Centralized, Localized, and Security Policy.

delete_data_prefix_list (*self, listid*)

Delete a Data Prefix List from vManage.

Parameters *listid* (*str*) – vManaged assigned list identifier

Returns Results from deletion attempt.

Return type response (dict)

get_data_prefix_list (*self*)

Get all Data Prefix Lists from vManage.

Returns

A list of all data prefix lists currently in vManage.

Return type response (dict)

get_policy_list_all (*self*)

Get all Policy Lists from vManage.

Returns

A list of all policy lists currently in vManage.

Return type response (dict)

post_data_prefix_list (*self, name, entries*)

Add a new Data Prefix List to vManage.

Parameters

- **name** (*str*) – name of the data prefix list
- **entries** (*list*) – a list of prefixes to add to the list

Returns

Results from attempting to add a new data prefix list.

Return type response (dict)

put_data_prefix_list (*self, name, listid, entries*)

Update an existing Data Prefix List on vManage.

Parameters

- **name** (*str*) – name of the data prefix list
- **listid** (*str*) – vManaged assigned list identifier
- **entries** (*list*) – a list of prefixes to add to the list

Returns

Results from attempting to update an existing data prefix list.

Return type response (dict)

delete_policy_list (*self*, *listType*, *listId*)

Deletes the specified policy list type

Parameters

- **listType** (*str*) – Policy list type
- **listId** (*str*) – ID of the policy list

Returns All data associated with a response.

Return type result (dict)

clear_policy_list_cache (*self*)

get_policy_list_list (*self*, *policy_list_type*='all', *cache*=True)

Get a list of policy lists

Parameters

- **policy_list_type** (*str*) – Policy list type (default: all)
- **cache** (*bool*) – Use cached data

Returns All data associated with a response.

Return type result (dict)

get_policy_list_dict (*self*, *policy_list_type*='all', *key_name*='name', *remove_key*=False, *cache*=True)

Get a dictionary of policy lists

Parameters

- **policy_list_type** (*str*) – Policy list type
- **key_name** (*string*) – The name of the attribute to use as the dictionary key
- **remove_key** (*boolean*) – Remove the search key from the element
- **cache** (*boolean*) – Use cached data

Returns All data associated with a response.

Return type result (dict)

get_policy_list_by_name (*self*, *policy_list_name*, *policy_list_type*='all')

Get a policy list by name

Parameters

- **policy_list_name** (*str*) – Policy list name
- **policy_list_type** (*str*) – Policy list type (default: all)

Returns All data associated with a response.

Return type result (dict)

get_policy_list_by_id (*self*, *policy_list_id*, *policy_list_type*='all')

Get a policy list by ID

Parameters

- **policy_list_name** (*str*) – Policy list ID
- **policy_list_type** (*str*) – Policy list type (default: all)

Returns All data associated with a response.

Return type result (dict)

add_policy_list (*self, policy_list*)

Add a new Policy List to vManage.

Parameters **policy_list** (*dict*) – The Policy List

Returns

Results from attempting to add a new prefix list.

Return type response (dict)

update_policy_list (*self, policy_list*)

Update an existing Policy List on vManage.

Parameters **policy_list** (*dict*) – The Policy List

Returns

Results from attempting to update an existing data prefix list.

Return type response (dict)

api.policy_updates

Cisco vManage Policy Lists API Methods.

Module Contents

class **api.policy_updates.PolicyUpdates** (*session, host, port=443*)

Bases: object

vManage Policy Updates API

Responsible for POST and PUT methods against vManage Policy APIs used in Centralized, Localized, and Security Policy.

get_device_ids (*self, template_id*)

GET vSmart device ids from vManage.

Parameters **templateid** (*str*) – vManaged assigned template identifier.

Returns device ids list.

Return type response (list)

get_device_inputs (*self, template_id, device_ids*)

GET vSmart device inputs from vManage.

Parameters

- **templateid** (*str*) – vManage assigned template identifier.
- **device_ids** (*list*) – vManage assigned device ids for vSmarts.

Returns device inputs.

Return type response (dict)

get_policy_id(*self, policy_type, policy_name*)

GET vSmart device ids from vManage.

Parameters

- **policy_type** (*str*) – vManage policy type i.e. approute, data, hubandspoke etc.
- **policy_name** (*str*) – policy name for user needs policy id.

Returns vManage assigned policy id.

Return type response (str)

get_policy_definition(*self, policy_type, policy_id*)

GET vSmart device ids from vManage.

Parameters

- **policy_type** (*str*) – vManage policy type i.e. approute, data, hubandspoke etc.
- **policy_id** (*str*) – vManage assigned policy id.

Returns policy definition.

Return type response (dict)

update_policy_definition(*self, policy_type, name, policy_id, policy_def, new_color, seq_name=None*)

update policy definition

Parameters

- **policy_type** (*str*) – vManage policy type i.e. approute, data, hubandspoke etc.
- **policy_id** (*str*) – vManage assigned policy id.
- **policy_def** (*dict*) – dict of data for policy updates to be done

Returns policy definition.

Return type response (dict)

`api.security_policy`

Cisco vManage Security Policy API Methods.

Module Contents

class `api.security_policy.SecurityPolicy` (*session, host, port=443*)

Bases: `object`

vManage Security Policy API

Responsible for DELETE, GET, POST, PUT methods against vManage Security Policy.

add_security_policy (*self, policy*)

Add a Security Policy from vManage.

Parameters **policy** – The Security Policy

Returns All data associated with a response.

Return type result (dict)

update_security_policy (*self, policy, policy_id*)

Update a Security from vManage.

Parameters

- **policy** – The Security Policy
- **policy_id** – The ID of the Security Policy to update

Returns All data associated with a response.

Return type result (dict)

delete_security_policy (*self, policyId*)

Deletes the specified security policy

Parameters **policyId** (*str*) – ID of the active security policy

Returns All data associated with a response.

Return type result (dict)

delete_security_definition (*self, definition, definitionId*)

Deletes the specified policy definition which include: 'zonebasedfw', 'urlfiltering', 'dnssecurity', 'intrusionprevention', 'advancedMalwareProtection' for 18.4.0 or greater and 'zonebasedfw' for

Parameters

- **definition** (*str*) – One of the above policy types
- **definitionId** (*str*) – ID of the policy definitions

Returns All data associated with a response.

Return type result (dict)

get_security_policy (*self*)

Obtain a list of all configured security policies

Returns All data associated with a response.

Return type result (dict)

get_security_definition (*self, definition*)

Obtain a list of various security definitions which include: 'zonebasedfw', 'urlfiltering', 'intrusionprevention', 'advancedMalwareProtection', 'dnssecurity'

Parameters **definition** (*str*) – One of the above policy types

Returns All data associated with a response.

Return type result (dict)

get_security_policy_list (*self*)

Get all Security Policies from vManage.

Returns

A list of all policy lists currently in vManage.

Return type response (list)

get_security_policy_dict (*self, key_name='policyName', remove_key=False*)

Get all Security Policies from vManage.

Parameters

- **key_name** (*str*) – The name of the attribute to use as the key

- **remove_key** (*bool*) – Remove the key from the dict (default: False)

Returns

A dict of all Security Policies currently in vManage.

Return type response (dict)

api.settings

Cisco vManage Settings API Methods.

Module Contents

class `api.settings.Settings` (*session, host, port=443*)

Bases: `object`

vManage Settings API

Responsible for DELETE, GET, POST, PUT methods against vManage Settings.

get_vmanage_org (*self*)

Get vManage organization

Args:

Returns The vManage organization.

Return type org (str)

set_vmanage_org (*self, org*)

Set vManage organization

Parameters **org** (*str*) – The organization name to set.

Returns The result of the POST operation

Return type result (dict)

get_vmanage_vbond (*self*)

Get vBond

Args:

Returns The result of the GET operation.

Return type result (dict)

set_vmanage_vbond (*self, vbond, vbond_port='12346'*)

Set vBond

Parameters

- **vbond** (*str*) – The vBond IP address.
- **vbond_port** (*str*) – The vBond port.

Returns The result of the POST operation.

Return type result (dict)

get_vmanage_ca_type (*self*)

Get vManage CA type

Args:

Returns The CA type.

Return type certificateSigning (str)

set_vmanage_ca_type (*self*, *ca_type*)

Set vManage CA type

Parameters *ca_type* (*str*) – The CA type.

Returns The result of the POST operation.

Return type result (dict)

set_vmanage_root_cert (*self*, *cert*, *timeout=20*)

Set vManage root certificate

Parameters

- **cert** (*str*) – The root certificate.
- **timeout** (*int*) – Timeout in seconds.

Returns:

get_vmanage_banner (*self*)

Get vManage Banner

Args:

Returns The banner for vmanage.

Return type bannerDetail (str)

api.utilities

Cisco vManage Utilities API Methods.

Module Contents

class api.utilities.Utilities (*session*, *host*, *port=443*)

Bases: object

Access to Various vManage Utilitiesinstance.

vManage has several utilities that are needed for correct execution of applications against the API. For example, this includes waiting for an action to complete before moving onto the next task.

get_active_count (*self*)

Provides number of active tasks on vManage.

Returns All data associated with a response.

Return type result (dict)

get_vmanage_version (*self*)

waitfor_action_completion (*self*, *action_id*)

upload_file (*self*, *input_file*)

Upload a file to vManage.

Parameters *input_file* (*str*) – The name of the file to upload.

Returns The status of the file upload.

Return type upload_status (str)

1.4.2 Package Contents

```
api.name = vmanage.api
```


INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

a

- `api`, 11
 - `api.authentication`, 11
 - `api.central_policy`, 12
 - `api.certificate`, 13
 - `api.cluster`, 14
 - `api.device`, 15
 - `api.device_templates`, 18
 - `api.feature_templates`, 21
 - `api.http_methods`, 22
 - `api.local_policy`, 23
 - `api.monitor_network`, 24
 - `api.policy_definitions`, 40
 - `api.policy_lists`, 42
 - `api.policy_updates`, 44
 - `api.security_policy`, 45
 - `api.settings`, 47
 - `api.utilities`, 48
- `apps`, 9
 - `apps.clean`, 9
 - `apps.files`, 10

d

- `data`, 3
 - `data.parse_methods`, 3
 - `data.policy_data`, 4
 - `data.template_data`, 7

u

- `utils`, 3

Symbols

`_get_device_type()`
 (*api.monitor_network.MonitorNetwork*
 method), 24

`_ipsec_query_params()`
 (*api.monitor_network.MonitorNetwork*
 method), 24

A

`activate_central_policy()`
 (*api.central_policy.CentralPolicy* *method*),
 12

`active_count_delay()`
 (*apps.clean.CleanVmanage* *method*), 9

`add_central_policy()`
 (*api.central_policy.CentralPolicy* *method*),
 12

`add_device_template()`
 (*api.device_templates.DeviceTemplates*
 method), 19

`add_feature_template()`
 (*api.feature_templates.FeatureTemplates*
 method), 21

`add_local_policy()` (*api.local_policy.LocalPolicy*
 method), 23

`add_policy_definition()`
 (*api.policy_definitions.PolicyDefinitions*
 method), 40

`add_policy_list()` (*api.policy_lists.PolicyLists*
 method), 44

`add_security_policy()`
 (*api.security_policy.SecurityPolicy* *method*), 45

`all_definition_types` (in *module*
 api.policy_definitions), 40

`api` (*module*), 11

`api.authentication` (*module*), 11

`api.central_policy` (*module*), 12

`api.certificate` (*module*), 13

`api.cluster` (*module*), 14

`api.device` (*module*), 15

`api.device_templates` (*module*), 18

`api.feature_templates` (*module*), 21

`api.http_methods` (*module*), 22

`api.local_policy` (*module*), 23

`api.monitor_network` (*module*), 24

`api.policy_definitions` (*module*), 40

`api.policy_lists` (*module*), 42

`api.policy_updates` (*module*), 44

`api.security_policy` (*module*), 45

`api.settings` (*module*), 47

`api.utilities` (*module*), 48

`apps` (*module*), 9

`apps.clean` (*module*), 9

`apps.files` (*module*), 10

`attach_to_template()`
 (*api.device_templates.DeviceTemplates*
 method), 19

Authentication (*class in api.authentication*), 11

C

`CentralPolicy` (*class in api.central_policy*), 12

`Certificate` (*class in api.certificate*), 13

`clean_all()` (*apps.clean.CleanVmanage* *method*), 9

`clean_central_policy()`
 (*apps.clean.CleanVmanage* *method*), 9

`clean_controller_attachments()`
 (*apps.clean.CleanVmanage* *method*), 9

`clean_device_templates()`
 (*apps.clean.CleanVmanage* *method*), 9

`clean_feature_templates()`
 (*apps.clean.CleanVmanage* *method*), 9

`clean_local_policy()`
 (*apps.clean.CleanVmanage* *method*), 9

`clean_policy_definitions()`
 (*apps.clean.CleanVmanage* *method*), 9

`clean_policy_lists()`
 (*apps.clean.CleanVmanage* *method*), 9

`clean_security_policy()`
 (*apps.clean.CleanVmanage* *method*), 9

`clean_vedge()` (*apps.clean.CleanVmanage* *method*),
 9

`clean_vedge_attachments()`
 (*apps.clean.CleanVmanage* *method*), 9

`CleanVmanage` (*class in apps.clean*), 9

clear_policy_list_cache()
 (*api.policy_lists.PolicyLists method*), 43
 Cluster (*class in api.cluster*), 14
 convert_definition_id_to_name()
 (*data.policy_data.PolicyData method*), 5
 convert_definition_name_to_id()
 (*data.policy_data.PolicyData method*), 5
 convert_device_template_to_id()
 (*data.template_data.TemplateData method*), 7
 convert_device_template_to_name()
 (*data.template_data.TemplateData method*), 7
 convert_list_id_to_name()
 (*data.policy_data.PolicyData method*), 5
 convert_list_name_to_id()
 (*data.policy_data.PolicyData method*), 5
 convert_policy_definition_to_id()
 (*data.policy_data.PolicyData method*), 5
 convert_policy_definition_to_name()
 (*data.policy_data.PolicyData method*), 5
 convert_policy_to_id()
 (*data.policy_data.PolicyData method*), 6
 convert_policy_to_name()
 (*data.policy_data.PolicyData method*), 5
 convert_sequences_to_id()
 (*data.policy_data.PolicyData method*), 5

D

data (*module*), 3
 data.parse_methods (*module*), 3
 data.policy_data (*module*), 4
 data.template_data (*module*), 7
 deactivate_central_policy()
 (*api.central_policy.CentralPolicy method*), 12
 definition_types_19_2_0 (in *module api.policy_definitions*), 40
 definition_types_19_3_0 (in *module api.policy_definitions*), 40
 delete_central_policy()
 (*api.central_policy.CentralPolicy method*), 12
 delete_data_prefix_list()
 (*api.policy_lists.PolicyLists method*), 42
 delete_device_template()
 (*api.device_templates.DeviceTemplates method*), 18
 delete_feature_template()
 (*api.feature_templates.FeatureTemplates method*), 21
 delete_local_policy()
 (*api.local_policy.LocalPolicy method*), 23
 delete_policy_definition()
 (*api.policy_definitions.PolicyDefinitions method*), 40

delete_policy_list()
 (*api.policy_lists.PolicyLists method*), 43
 delete_security_definition()
 (*api.security_policy.SecurityPolicy method*), 46
 delete_security_policy()
 (*api.security_policy.SecurityPolicy method*), 46
 detach_from_template()
 (*api.device_templates.DeviceTemplates method*), 20
 Device (*class in api.device*), 15
 DeviceTemplates (*class in api.device_templates*), 18

E

export_attachments_to_file()
 (*apps.files.Files method*), 10
 export_central_policy_list()
 (*data.policy_data.PolicyData method*), 6
 export_device_template_list()
 (*data.template_data.TemplateData method*), 7
 export_local_policy_list()
 (*data.policy_data.PolicyData method*), 6
 export_policy_definition_list()
 (*data.policy_data.PolicyData method*), 5
 export_policy_to_file() (*apps.files.Files method*), 10
 export_security_policy_list()
 (*data.policy_data.PolicyData method*), 6
 export_templates_to_file() (*apps.files.Files method*), 10

F

FeatureTemplates (*class in api.feature_templates*), 21
 Files (*class in apps.files*), 10

G

generalTemplates_to_id()
 (*data.template_data.TemplateData method*), 7
 generate_csr() (*api.certificate.Certificate method*), 13
 get_aaa_users() (*api.monitor_network.MonitorNetwork method*), 24
 get_active_count() (*api.utilities.Utilities method*), 48
 get_arp_table() (*api.monitor_network.MonitorNetwork method*), 24
 get_attachments()
 (*api.device_templates.DeviceTemplates method*), 20
 get_bfd_device_state()
 (*api.monitor_network.MonitorNetwork method*), 25

<code>get_bfd_device_state_tloc()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 25	<code>get_central_policy_list()</code> (<i>api.central_policy.CentralPolicy</i> <i>method</i>), 13
<code>get_bfd_history()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 24	<code>get_cluster_connected_devices_list()</code> (<i>api.cluster.Cluster</i> <i>method</i>), 14
<code>get_bfd_links()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 25	<code>get_cluster_health_details_list()</code> (<i>api.cluster.Cluster</i> <i>method</i>), 14
<code>get_bfd_sessions()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 25	<code>get_cluster_health_status_list()</code> (<i>api.cluster.Cluster</i> <i>method</i>), 14
<code>get_bfd_summary()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 25	<code>get_cluster_ip_addresses_dict()</code> (<i>api.cluster.Cluster</i> <i>method</i>), 14
<code>get_bfd_tloc()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 25	<code>get_cluster_list()</code> (<i>api.cluster.Cluster</i> <i>method</i>), 14
<code>get_bgp_neighbors()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 25	<code>get_cluster_node_properties()</code> (<i>api.cluster.Cluster</i> <i>method</i>), 15
<code>get_bgp_routes()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 26	<code>get_cluster_ready_state()</code> (<i>api.cluster.Cluster</i> <i>method</i>), 15
<code>get_bgp_summary()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 26	<code>get_cluster_tenancy_mode()</code> (<i>api.cluster.Cluster</i> <i>method</i>), 15
<code>get_cellular_connections()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 26	<code>get_cluster_vmanage_details_list()</code> (<i>api.cluster.Cluster</i> <i>method</i>), 15
<code>get_cellular_hardware()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 26	<code>get_control_affinity_config()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 27
<code>get_cellular_modems()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 26	<code>get_control_affinity_status()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 27
<code>get_cellular_networks()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 26	<code>get_control_connections()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 27
<code>get_cellular_profiles()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 27	<code>get_control_connections_history()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 28
<code>get_cellular_radios()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 27	<code>get_control_count()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 28
<code>get_cellular_sessions()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 27	<code>get_control_links()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 28
<code>get_cellular_status()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 27	<code>get_control_local_properties()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 28
<code>get_central_policy()</code> (<i>api.central_policy.CentralPolicy</i> <i>method</i>), 13	<code>get_control_summary()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 28
<code>get_central_policy_dict()</code> (<i>api.central_policy.CentralPolicy</i> <i>method</i>), 13	<code>get_control_valid_devices()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 29
	<code>get_control_valid_vsmarts()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 29
	<code>get_control_wan_interface()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 29

<code>get_control_wan_interface_color()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 29	<code>(api.monitor_network.MonitorNetwork</code> <i>method</i>), 29
<code>get_data_prefix_list()</code> (<i>api.policy_lists.PolicyLists</i> <i>method</i>), 42	<code>get_dhcp_servers()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 30
<code>get_definition_types()</code> (<i>api.policy_definitions.PolicyDefinitions</i> <i>method</i>), 40	<code>get_dot1x_clients()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 30
<code>get_device_config()</code> (<i>api.device.Device</i> <i>method</i>), 16	<code>get_dot1x_interfaces()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 30
<code>get_device_config_dict()</code> (<i>api.device.Device</i> <i>method</i>), 16	<code>get_dot1x_radius()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 30
<code>get_device_config_list()</code> (<i>api.device.Device</i> <i>method</i>), 16	<code>get_dpi_applications()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 30
<code>get_device_data()</code> (<i>api.device.Device</i> <i>method</i>), 17	<code>get_dpi_common_applications()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 30
<code>get_device_ids()</code> (<i>api.policy_updates.PolicyUpdates</i> <i>method</i>), 44	<code>get_dpi_device_fields()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 30
<code>get_device_inputs()</code> (<i>api.policy_updates.PolicyUpdates</i> <i>method</i>), 44	<code>get_dpi_device_fields_details()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 31
<code>get_device_list()</code> (<i>api.device.Device</i> <i>method</i>), 15	<code>get_dpi_flows()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 31
<code>get_device_running_config()</code> (<i>api.device_templates.DeviceTemplates</i> <i>method</i>), 20	<code>get_dpi_summary()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 31
<code>get_device_status()</code> (<i>api.device.Device</i> <i>method</i>), 16	<code>get_dpi_supported_applications()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 31
<code>get_device_status()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 29	<code>get_eigrp_interfaces()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 31
<code>get_device_status_dict()</code> (<i>api.device.Device</i> <i>method</i>), 16	<code>get_eigrp_routes()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 31
<code>get_device_status_list()</code> (<i>api.device.Device</i> <i>method</i>), 16	<code>get_eigrp_topology()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 32
<code>get_device_system_info()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 29	<code>get_feature_template_dict()</code> (<i>api.feature_templates.FeatureTemplates</i> <i>method</i>), 21
<code>get_device_template_dict()</code> (<i>api.device_templates.DeviceTemplates</i> <i>method</i>), 18	<code>get_feature_template_list()</code> (<i>api.feature_templates.FeatureTemplates</i> <i>method</i>), 21
<code>get_device_template_list()</code> (<i>api.device_templates.DeviceTemplates</i> <i>method</i>), 18	<code>get_feature_templates()</code> (<i>api.feature_templates.FeatureTemplates</i> <i>method</i>), 21
<code>get_device_template_object()</code> (<i>api.device_templates.DeviceTemplates</i> <i>method</i>), 18	<code>get_hardware_alarms()</code> (<i>api.monitor_network.MonitorNetwork</i>
<code>get_device_templates()</code> (<i>api.device_templates.DeviceTemplates</i> <i>method</i>), 18	
<code>get_device_templates_for_feature()</code> (<i>api.feature_templates.FeatureTemplates</i> <i>method</i>), 22	
<code>get_dhcp_clients()</code> (<i>api.monitor_network.MonitorNetwork</i> <i>method</i>), 29	
<code>get_dhcp_interfaces()</code>	

`method`), 32
`get_hardware_environment()`
 (*api.monitor_network.MonitorNetwork*
 method), 32
`get_hardware_inventory()`
 (*api.monitor_network.MonitorNetwork*
 method), 32
`get_hardware_status_summary()`
 (*api.monitor_network.MonitorNetwork*
 method), 32
`get_hardware_system()`
 (*api.monitor_network.MonitorNetwork*
 method), 32
`get_igmp_groups()`
 (*api.monitor_network.MonitorNetwork*
 method), 32
`get_igmp_interfaces()`
 (*api.monitor_network.MonitorNetwork*
 method), 32
`get_igmp_statistics()`
 (*api.monitor_network.MonitorNetwork*
 method), 33
`get_igmp_summary()`
 (*api.monitor_network.MonitorNetwork*
 method), 33
`get_interface_vpn()`
 (*api.monitor_network.MonitorNetwork*
 method), 33
`get_ip_fib()` (*api.monitor_network.MonitorNetwork*
 method), 33
`get_ip_nat64_translations()`
 (*api.monitor_network.MonitorNetwork*
 method), 34
`get_ip_nat_translations()`
 (*api.monitor_network.MonitorNetwork*
 method), 34
`get_ip_route_table()`
 (*api.monitor_network.MonitorNetwork*
 method), 33
`get_ipsec_inbound_connections()`
 (*api.monitor_network.MonitorNetwork*
 method), 34
`get_ipsec_localsa()`
 (*api.monitor_network.MonitorNetwork*
 method), 34
`get_ipsec_outbound_connections()`
 (*api.monitor_network.MonitorNetwork*
 method), 34
`get_ipsec_pwk_inbound_connections()`
 (*api.monitor_network.MonitorNetwork*
 method), 34
`get_ipsec_pwk_localsa()`
 (*api.monitor_network.MonitorNetwork*
 method), 35
`get_ipsec_pwk_outbound_connections()`
 (*api.monitor_network.MonitorNetwork*
 method), 35
`get_local_policy()` (*api.local_policy.LocalPolicy*
 method), 23
`get_local_policy_dict()`
 (*api.local_policy.LocalPolicy method*), 24
`get_local_policy_list()`
 (*api.local_policy.LocalPolicy method*), 23
`get_multi_attach_payload()`
 (*api.device_templates.DeviceTemplates*
 method), 20
`get_multicast_replicator()`
 (*api.monitor_network.MonitorNetwork*
 method), 35
`get_multicast_rpf()`
 (*api.monitor_network.MonitorNetwork*
 method), 35
`get_multicast_topology()`
 (*api.monitor_network.MonitorNetwork*
 method), 35
`get_multicast_tunnel()`
 (*api.monitor_network.MonitorNetwork*
 method), 36
`get_omp_peers()` (*api.monitor_network.MonitorNetwork*
 method), 36
`get_omp_routes_advertised()`
 (*api.monitor_network.MonitorNetwork*
 method), 36
`get_omp_routes_received()`
 (*api.monitor_network.MonitorNetwork*
 method), 36
`get_omp_summary()`
 (*api.monitor_network.MonitorNetwork*
 method), 36
`get_orchestrator_connections()`
 (*api.monitor_network.MonitorNetwork*
 method), 37
`get_orchestrator_connections_history()`
 (*api.monitor_network.MonitorNetwork*
 method), 37
`get_orchestrator_local_properties()`
 (*api.monitor_network.MonitorNetwork*
 method), 37
`get_orchestrator_summary()`
 (*api.monitor_network.MonitorNetwork*
 method), 36
`get_ospf_database()`
 (*api.monitor_network.MonitorNetwork*
 method), 37
`get_ospf_database_external()`
 (*api.monitor_network.MonitorNetwork*
 method), 38
`get_ospf_database_summary()`


```

        (api.monitor_network.MonitorNetwork
         method), 37
get_ospf_interfaces()
    (api.monitor_network.MonitorNetwork
     method), 37
get_ospf_neighbors()
    (api.monitor_network.MonitorNetwork
     method), 37
get_ospf_process()
    (api.monitor_network.MonitorNetwork
     method), 38
get_ospf_routes()
    (api.monitor_network.MonitorNetwork
     method), 37
get_pim_interfaces()
    (api.monitor_network.MonitorNetwork
     method), 38
get_pim_neighbors()
    (api.monitor_network.MonitorNetwork
     method), 38
get_pim_rp_mapping()
    (api.monitor_network.MonitorNetwork
     method), 38
get_pim_statistics()
    (api.monitor_network.MonitorNetwork
     method), 38
get_policy_definition()
    (api.policy_definitions.PolicyDefinitions
     method), 41
get_policy_definition()
    (api.policy_updates.PolicyUpdates method), 45
get_policy_definition_dict()
    (api.policy_definitions.PolicyDefinitions
     method), 41
get_policy_definition_list()
    (api.policy_definitions.PolicyDefinitions
     method), 41
get_policy_id() (api.policy_updates.PolicyUpdates
                 method), 44
get_policy_list_all()
    (api.policy_lists.PolicyLists method), 42
get_policy_list_by_id()
    (api.policy_lists.PolicyLists method), 43
get_policy_list_by_name()
    (api.policy_lists.PolicyLists method), 43
get_policy_list_dict()
    (api.policy_lists.PolicyLists method), 43
get_policy_list_list()
    (api.policy_lists.PolicyLists method), 43
get_security_definition()
    (api.security_policy.SecurityPolicy method), 46
get_security_information()
    (api.monitor_network.MonitorNetwork
     method), 38
get_security_policy()
    (api.security_policy.SecurityPolicy method), 46
get_security_policy_dict()
    (api.security_policy.SecurityPolicy method), 46
get_security_policy_list()
    (api.security_policy.SecurityPolicy method), 46
get_software() (api.monitor_network.MonitorNetwork
               method), 39
get_template_attachments()
    (api.device_templates.DeviceTemplates
     method), 19
get_template_input()
    (api.device_templates.DeviceTemplates
     method), 19
get_umbrella_sig_tunnels()
    (api.monitor_network.MonitorNetwork
     method), 39
get_users() (api.monitor_network.MonitorNetwork
            method), 39
get_users_list() (api.monitor_network.MonitorNetwork
                 method), 39
get_vmanage_banner() (api.settings.Settings
                     method), 48
get_vmanage_ca_type() (api.settings.Settings
                      method), 47
get_vmanage_org() (api.settings.Settings method),
                  47
get_vmanage_root_cert()
    (api.certificate.Certificate method), 14
get_vmanage_vbond() (api.settings.Settings
                    method), 47
get_vmanage_version() (api.utilities.Utilities
                      method), 48
get_vrrp() (api.monitor_network.MonitorNetwork
           method), 39
get_wlan_clients()
    (api.monitor_network.MonitorNetwork
     method), 39
get_wlan_interfaces()
    (api.monitor_network.MonitorNetwork
     method), 39
get_wlan_radios()
    (api.monitor_network.MonitorNetwork
     method), 39
get_wlan_radius()
    (api.monitor_network.MonitorNetwork
     method), 40
get_zscaler_sig_tunnels()
    (api.monitor_network.MonitorNetwork
     method), 40

```

H

HttpMethods (class in *api.http_methods*), 22

I

import_attachment_list() *(data.template_data.TemplateData method)*, 8
 import_attachments_from_file() *(apps.files.Files method)*, 10
 import_central_policy_list() *(data.policy_data.PolicyData method)*, 6
 import_device_template_list() *(data.template_data.TemplateData method)*, 8
 import_feature_template_list() *(data.template_data.TemplateData method)*, 7
 import_local_policy_list() *(data.policy_data.PolicyData method)*, 6
 import_policy_definition_list() *(data.policy_data.PolicyData method)*, 5
 import_policy_from_file() *(apps.files.Files method)*, 10
 import_policy_list_list() *(data.policy_data.PolicyData method)*, 4
 import_security_policy_list() *(data.policy_data.PolicyData method)*, 6
 import_templates_from_file() *(apps.files.Files method)*, 10
 install_device_cert() *(api.certificate.Certificate method)*, 13

L

list_to_dict() *(in module utils)*, 3
 LocalPolicy *(class in api.local_policy)*, 23
 login() *(api.authentication.Authentication method)*, 11

M

MonitorNetwork *(class in api.monitor_network)*, 24

N

name *(in module api)*, 49
 name *(in module apps)*, 11
 name *(in module data)*, 9

P

parse_config() *(data.parse_methods.ParseMethods static method)*, 4
 parse_data() *(data.parse_methods.ParseMethods static method)*, 3
 parse_id() *(data.parse_methods.ParseMethods static method)*, 4
 parse_status() *(data.parse_methods.ParseMethods static method)*, 4
 ParseMethods *(class in data.parse_methods)*, 3
 PolicyData *(class in data.policy_data)*, 4
 PolicyDefinitions *(class in api.policy_definitions)*, 40

PolicyLists *(class in api.policy_lists)*, 42
 PolicyUpdates *(class in api.policy_updates)*, 44
 post_data_prefix_list() *(api.policy_lists.PolicyLists method)*, 42
 post_device() *(api.device.Device method)*, 17
 post_device_cli_mode() *(api.device.Device method)*, 16
 post_reset_interface() *(api.device.Device method)*, 17
 push_certificates() *(api.certificate.Certificate method)*, 13
 put_data_prefix_list() *(api.policy_lists.PolicyLists method)*, 42
 put_device_decommission() *(api.device.Device method)*, 17

R

reactivate_central_policy() *(api.central_policy.CentralPolicy method)*, 12
 reattach_device_template() *(api.device_templates.DeviceTemplates method)*, 19
 reattach_multi_device_templates() *(api.device_templates.DeviceTemplates method)*, 20
 request() *(api.http_methods.HttpMethods method)*, 22

S

SecurityPolicy *(class in api.security_policy)*, 45
 set_vmanage_ca_type() *(api.settings.Settings method)*, 48
 set_vmanage_org() *(api.settings.Settings method)*, 47
 set_vmanage_root_cert() *(api.settings.Settings method)*, 48
 set_vmanage_vbond() *(api.settings.Settings method)*, 47
 Settings *(class in api.settings)*, 47
 STANDARD_HEADERS *(in module api.http_methods)*, 22
 STANDARD_TIMEOUT *(in module api.http_methods)*, 22
 subTemplates_to_id() *(data.template_data.TemplateData method)*, 8
 subTemplates_to_name() *(data.template_data.TemplateData method)*, 8

T

TemplateData *(class in data.template_data)*, 7

U

update_central_policy() *(api.central_policy.CentralPolicy method)*, 12

`update_device_template()`
 (*api.device_templates.DeviceTemplates*
 method), 19

`update_feature_template()`
 (*api.feature_templates.FeatureTemplates*
 method), 21

`update_local_policy()`
 (*api.local_policy.LocalPolicy method*), 23

`update_policy_definition()`
 (*api.policy_definitions.PolicyDefinitions*
 method), 41

`update_policy_definition()`
 (*api.policy_updates.PolicyUpdates method*), 45

`update_policy_list()`
 (*api.policy_lists.PolicyLists method*), 44

`update_security_policy()`
 (*api.security_policy.SecurityPolicy method*), 45

`upload_file()` (*api.utilities.Utilities method*), 48

`Utilities` (*class in api.utilities*), 48

`utils` (*module*), 3

V

`VALID_STATUS_CODES` (*in module api.http_methods*),
22

`VALID_STATUS_CODES` (in module
 data.parse_methods), 3

W

`waitfor_action_completion()`
 (*api.utilities.Utilities method*), 48