Python vManage

Release 0.2.3

Cisco Public Sector

May 07, 2020

CONTENTS:

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

CHAPTER

ONE

API REFERENCE

This page contains auto-generated API reference documentation[1].

1.1 utils

1.1.1 Module Contents

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

1.2 __main__

1.2.1 Module Contents

class __main__.**CatchAllExceptions**(*name=None*, *commands=None*, ***attrs*)

Bases: click.Group

__call__(*self*, **args*, ***kwargs*)

class __main__.**Viptela**(*host*, *username*, *password*)

Bases: object

property auth(*self*)

__main__.**vmanage**(*ctx*, *host*, *username*, *password*)

1.3 api

1.3.1 Submodules

api.authentication

Cisco vManage Authentication API Methods.

[1] Created with sphinx-autoapi

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 result (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_list(*self*)
Get all Central Policies from vManage.

Returns

A list of all policy lists currently in vManage.

Return type response (dict)

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

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

waitfor_action_completion(*self*, *action_id*)

api.centralized_policy

Cisco vManage Centralized Policy API Methods.

Module Contents

class api.centralized_policy.**CentralizedPolicy**(*session*, *host*, *port=443*)
Bases: object

vManage Centralized Policy API

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

deactivate_centralized_policy(*self*, *policyId*)
Deactivates the current active centralized policy

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

Returns All data associated with a response.

Return type result (dict)

delete_centralized_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)

delete_policy_definition (*self*, *definition*, *definitionId*)
Deletes the specified policy definition which include: 'control','mesh','hubandspoke','vpnmembershipgroup', 'approute','data','cflowd'

Parameters

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

Returns All data associated with a response.

Return type result (dict)

get_centralized_policy (*self*)
Obtain a list of all configured centralized policies

Returns All data associated with a response.

Return type result (dict)

get_policy_definition (*self*, *definition*)
Obtain a list of various policy definitions which include: 'control','mesh','hubandspoke','vpnmembershipgroup', 'approute','data','cflowd'

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

Returns All data associated with a response.

Return type result (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*)

install_device_cert (*self*, *cert*)

push_certificates (*self*)
Push certificates to all controllers

Returns The action ID of the push command.

Return type result (str)

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

get_device_config_list(*self*, *device_type*)
Get the config status of a list of devices

Parameters **device_type** (*str*) – vedge or controller

Returns All data associated with a response.

Return type result (dict)

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

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)

api.device_inventory

Cisco vManage Device Inventory API Methods.

Module Contents

class api.device_inventory.**DeviceInventory** (*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*, *deviceIP*, *deviceType*)
Update a device to CLI mode

Parameters

- **deviceId** (*str*) – uuid for device object
- **deviceIP** (*str*) – system IP equivalent
- **deviceType** (*str*) – vedge or controller

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.

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 retreive.

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

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

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)

import_device_template_list (*self*, *device_template_list*, *check_mode=False*, *update=False*)
Import a list of feature templates to vManage.

Parameters

- **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

- **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)

attach_to_template (*self*, *template_id*, *uuid*, *system_ip*, *host_name*, *site_id*, *variables*)
Attach and device to a template

Parameters

- **template_id** (*str*) – The template ID to attach to
- **uuid** (*str*) – The UUID of the device to attach
- **system_ip** (*str*) – The System IP of the system to attach

- **host_name** (*str*) – The host-name of the device to attach
- **variables** (*dict*) – The variables needed by the template

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
- **system_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)

generalTemplates_to_id(*self*, *generalTemplates*)

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*)
Add a feature template to 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*) – Wheter 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*) – 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

Returns All data associated with a response.

Return type result (dict)

import_feature_template_list(*self*, *feature_template_list*, *check_mode=False*, *update=False*)
Add a list of feature templates to vManage.

Parameters

- **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)

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*)

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_localized_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_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*)

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

api.localized_policy

Cisco vManage Localized Policy API Methods.

Module Contents

class api.localized_policy.**LocalizedPolicy**(*session*, *host*, *port=443*)
Bases: object

vManage Localized Policy API

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

delete_localized_definition(*self*, *definition*, *definitionId*)
Deletes the specified policy definition which include: 'qosmap','rewriterule','acl','aclv6','vedgeroute'

Parameters

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

Returns All data associated with a response.

Return type result (dict)

delete_localized_policy(*self*, *policy_id*)
Deletes the specified localized policy

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

Returns All data associated with a response.

Return type result (dict)

get_localized_definition(*self*, *definition*)
Obtain a list of various policy definitions which include: 'qosmap','rewriterule','acl','aclv6','vedgeroute'

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

Returns All data associated with a response.

Return type result (dict)

get_localized_policy(*self*)
Obtain a list of all configured localized policies

Returns All data associated with a response.

Return type result (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_control_connections(*self*, *system_ip*)
Provides current control connections for device.

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

Returns All data associated with a response.

Return type result (dict)

get_control_connections_history(*self*, *system_ip*)
Provides control connections history 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_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)

api.policy_definitions

Cisco vManage Policy Definitions API Methods.

Module Contents

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.

convert_list_id_to_name(*self*, *id_list*)

convert_list_name_to_id(*self*, *name_list*)

convert_definition_id_to_name(*self*, *policy_definition*)

convert_definition_name_to_id(*self*, *policy_definition*)

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 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)

update_policy_definition(*self*, *policy_definition*, *policy_definition_id*)
Update 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)

get_policy_definition(*self*, *definition_type*, *definition_id*)
Get 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)

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 retreive

Returns

A list of all definition lists currently in vManage.

Return type response (dict)

get_policy_definition_dict(*self*, *t*, *key_name='name'*, *remove_key=False*)

convert_sequences_to_id(*self*, *sequence_list*)

import_policy_definition_list(*self*, *policy_definition_list*, *update=False*, *push=False*, *check_mode=False*, *force=False*)

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
- **cache** (*bool*) – Whether to cache the response

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*)

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)

import_policy_list_list (*self*, *policy_list_list*, *push=False*, *update=False*, *check_mode=False*, *force=False*)
Import a list of policy lists into vManage

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)

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.

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 definition

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)

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(*self*)

Obtain a list of all configured security policies

Returns All data associated with a response.

Return type result (dict)

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)

1.3.2 Package Contents

api.**name = vmanage.api**

1.4 apps

1.4.1 Submodules

apps.files

Cisco vManage Files API Methods.

Module Contents

class apps.files.**Files**(*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.

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

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

export_policy_to_file(*self*, *export_file*)

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

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

import_attachments_from_file(*self*, *file*, *update=False*, *check_mode=False*, *name_list=None*, *template_type=None*)

apps.reset_vmanage

Reset vManage Application.

Module Contents

class apps.reset_vmanage.**ResetVmanage**(*session*, *host*, *port=443*)
Bases: object

Reset all configuratios on a vManage instance.

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

active_count_delay(*self*)

execute(*self*)

1.4.2 Package Contents

apps.**name = vmanage.apps**

1.5 data

1.5.1 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.

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)

1.5.2 Package Contents

data.**name = vmanage.data**

1.6 cli

1.6.1 Subpackages

cli.activate

Submodules

cli.activate.central_policy

Module Contents

cli.activate.central_policy.**central_policy**(*ctx*, *name*)
Activate Central Policy

Package Contents

cli.activate.**activate**()
Activate commands

cli.certificate

Package Contents

cli.certificate.**push**(*ctx*)
Push certificates to all controllers

cli.certificate.**certificate**()
Certficate commands

cli.deactivate

Submodules

cli.deactivate.central_policy

Module Contents

cli.deactivate.central_policy.**central_policy**(*ctx*, *name*, *policy_id*)
deactivate Central Policy

Package Contents

cli.deactivate.**deactivate**()
Deactivate commands

cli.export

Submodules

cli.export.attachments

Module Contents

cli.export.attachments.**attachments**(*ctx*, *device_type*, *name*, *output_file*)
Export attachments to file

cli.export.policies

Module Contents

cli.export.policies.**policies**(*ctx*, *export_file*)
Export policies to file

cli.export.templates

Module Contents

cli.export.templates.**templates**(*ctx*, *template_type*, *name*, *export_file*)
Export templates to file

Package Contents

cli.export.**export**()
Export commands

cli.import_cmd

Submodules

cli.import_cmd.attachments

Module Contents

cli.import_cmd.attachments.**attachments**(*ctx*, *input_file*, *check*, *update*, *name*, *template_type*)
Import attachments from file

cli.import_cmd.policies

Module Contents

cli.import_cmd.policies.**policies**(*ctx*, *input_file*, *check*, *update*, *push*, *diff*)
Import policies from file

cli.import_cmd.serial_file

Module Contents

cli.import_cmd.serial_file.**serial_file**(*ctx*, *input_file*)
Import serial file

cli.import_cmd.templates

Module Contents

cli.import_cmd.templates.**templates**(*ctx*, *input_file*, *check*, *update*, *diff*, *name*, *template_type*)
Import templates from file

Package Contents

cli.import_cmd.**import_cmd**()
Import commands

cli.show

Submodules

cli.show.control

Module Contents

cli.show.control.**connections**(*ctx*, *device*, *json*)
Show control connections

cli.show.control.**connections_history**(*ctx*, *device*, *json*)
Show control connections history

cli.show.control.**control**()
Show control information

cli.show.device

Module Contents

cli.show.device.**status**(*ctx*, *dev*, *device_type*, *json*)
Show device status information

cli.show.device.**config**(*ctx*, *dev*, *device_type*, *json*)
Show device config information

cli.show.device.**device**()
Show device information

cli.show.interface

Module Contents

cli.show.interface.**list_interface**(*ctx*, *device*, *json*)
Show Interfaces

cli.show.interface.**interface**()
Show real-time information

cli.show.omp

Module Contents

cli.show.omp.**peers**(*ctx*, *device*, *json*)
Show OMP peer information

cli.show.omp.**omp**()
Show OMP information

cli.show.policies

Module Contents

cli.show.policies.**list_cmd**(*ctx*, *name*, *json*, *policy_list_type*)
Show policy list information

cli.show.policies.**definition**(*ctx*, *name*, *json*, *definition_type*)
Show policy definition information

cli.show.policies.**central**(*ctx*, *name*, *json*)
Show central policy information

cli.show.policies.**local**(*ctx*, *name*, *json*)
Show local policy information

cli.show.policies.**policies**()
Show policy information

cli.show.route

Module Contents

cli.show.route.**table**(*ctx*, *device*, *json*)
Show Interfaces

cli.show.route.**route**()
Show device route information

cli.show.templates

Module Contents

cli.show.templates.**templates**(*ctx*, *template_type*, *diff*, *default*, *name*, *json*)
Show template information

Package Contents

cli.show.**show**()
Show commands

1.6.2 Package Contents

cli.**name = vmanage.cli**

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX

Symbols

A

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W