vManage\_functions.py

This document describes de code in vManage\_functions.py script:

'''

Example script to perform some vManage REST API calls through functions.

'''

This is a multiline comment just for documentation purposes.

# Import section

import requests

import sys

import json

from requests.packages.urllib3.exceptions import InsecureRequestWarning

requests.packages.urllib3.disable\_warnings(InsecureRequestWarning)

This is the import of the required libraries.

# Constants definition

VMANAGEIP = '198.18.1.10'

USERNAME = 'admin'

PASSWORD = 'admin'

This is the section of the constants definition.

The script is prepared to be run against the dCloud vManage instance. If we need to run the script against another vManage instance, we just need to change the values of the constants and set the IP address, username and password of the new vManage instance.

# Functions definition

def login(vmanageip, username, password):

# Function to perform a login to vManage

url = 'https://' + vmanageip + '/j\_security\_check'

data = {"j\_username" : username, "j\_password" : password}

sess = requests.session()

response = sess.post(url=url, data=data, verify=False)

if response.status\_code != 200:

print('Login Failed')

sys.exit(0)

return(sess)

This function makes a login to the vManage. It requires three parameters: vManage IP address, username and password.

If the login fails it prints a message (Login Failed) and ends the script.

If everything goes well, it returns the session (sess).

def get(vmanageip, apiurl):

# Function to perform a GET request to vManage

url = 'https://' + vmanageip + '/dataservice' + apiurl

response = sess.get(url=url, verify=False)

if response.status\_code != 200:

print('Get request failed\nStatus Code: ' + str(response.status\_code))

sys.exit(0)

return(response.content)

This function makes a GET request to the vManage. It requires two parameters: vManage IP address and the last part of the URL for the API call (the part that goes after /dataservice. This is what can be found in apidocs.

If the request fails it prints a message including the status code, and ends the script.

If everything goes well, it returns the GET request response (response.content).

# Main

# Login to vmanage

sess = login(VMANAGEIP, USERNAME, PASSWORD)

The first thing to do in the main program is to make a login to vManage.

# Make an API GET request for feature templates

featureTemplates = json.loads(get(VMANAGEIP, '/template/feature'))

print('\nList of Feature Templates: ')

print('%-60s%-60s' %('Template Name' , 'Template ID'))

for i in range (0,len(featureTemplates["data"])):

    print('%-60s%-60s'%(featureTemplates["data"][i]["templateName"] ,featureTemplates["data"][i]["templateId"]))

print()

print('Total number of Templates: ' , len(featureTemplates["data"])-1)

The second line of this part is the call to the get function. We request to make a GET request to /template/feature. The response.content is loaded in the featureTemplates variable in JSON format.

The rest of the lines are the processing of the received response to print it in the terminal in a readable way.

# Make an API GET request for device list

devices = json.loads(get(VMANAGEIP, '/device'))

print('\nList of devices:')

print('%-18s%-12s%-40s' % ('System IP', 'Site-ID', 'UUID'))

for i in range (0,len(devices["data"])):

    if "vedge" in devices["data"][i]["device-type"]:

        print('%-18s%-12s%-40s' % (devices["data"][i]["system-ip"], devices["data"][i]["site-id"], devices["data"][i]["uuid"]))

The second line of this part is the call to the get function. We request to make a GET request to /device. The response.content is loaded in the devices variable in JSON format.

The rest of the lines are the processing of the received response to print it in the terminal in a readable way.