calls.py and vManage.py

This document describes vManage.py library and the calls script. vManage.py is a library that implements the vManage class with login, get and post methods. This vManage class is imported in the calls.py script.

The vManage.py code is

'''

Library to import to facilitate calls to the vManage API

'''

This is just a comment.

# Import section

import requests

import sys

import json

from requests.packages.urllib3.exceptions import InsecureRequestWarning

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

These are the imports.

#Class definition

class vManage:

def \_\_init\_\_(self, vmanageip, username, password):

self.vmanageip = vmanageip

self.session = {}

self.login(self.vmanageip, username, password)

def login(self, 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)

self.session[vmanageip] = sess

def get(self, apiurl):

# Function to perform a GET request to vManage

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

response = self.session[self.vmanageip].get(url, verify=False)

if response.status\_code != 200:

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

sys.exit(0)

return(response.content)

def post(self, apiurl, payload, headers={'Content-Type': 'application/json'}):

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

payload = json.dumps(payload)

response = self.session[self.vmanage\_ip].post(url=url, data=payload, headers=headers, verify=False)

return(response.content)

The above is the class definition.

The class name is vManage, and it implements login, get and post methods.

To create and object of class vManage, three parameters are required: the vManage IP address, the username and the password.

The login method is called when an object of class vManage is created (login method is invoked by \_\_init\_\_.

The get method requires the apiurl parameter. This is the part of the URL for the GET request that goes after /datavervice.

For instance, to make a request for: <https://vmanageip/dataservice/device>, the value of apiurl must be /device.

To make a request for: <https://vmanageip/dataservice/template/feature> , the value of apiurl must be /template/feature.

The post method has not been tested.

def main(args):

if not len(args) == 3:

print(\_\_doc\_\_)

return

vmanageip, username, password = args[0], args[1], args[2]

myvManage = vManage(vmanageip, username, password)

#Example request to get devices from the vmanage "url=https://vmanageip/dataservice/device"

response = myvManage.get('/device')

print(response)

if \_\_name\_\_ == "\_\_main\_\_":

sys.exit(main(sys.argv[1:]))

This is the main part.

The calls.py code is:

'''

Make some calls to the vManage API

'''

This is a comment.

# Import section

from vManage import vManage

import json

This is the import section. We import the vManage class from the vManage library.

# 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.

# Main

myvManage = vManage(VMANAGEIP, USERNAME, PASSWORD)

# Make an API GET request for feature templates

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

print('\nList of Feature Templates: ')

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

for i in range (0,len(featureTemplates["data"])-1):

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

print('\nTotal number of Templates: ' , len(featureTemplates["data"]))

# Make an API GET request for device list

devices = json.loads(myvManage.get('/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"]))

This is the main part of the script.

First an object called myvManage of class vManage is created and then we make two GET requests to it.