

```

#!/usr/bin/env python3

# sample client which manipulates switch1 device
# through FishNet proxy

import requests
import sys
import inspect

from flask_api import status

# login/get token are global constants
USER = 'pyuser'
PASSWORD = 'cis3435!'
BASE_URL = 'http://127.0.0.1:5000/'

# log in to the proxy, returns token
def login(url):
    global USER, PASSWORD

    # pload is our payload for the POST
    pload = {'user':USER, 'password':PASSWORD}
    r = requests.post(url,data = pload)
    # print('STATUS CODE',r.status_code) # debug
    # print(r.text)
    return r.status_code, r.text

# returns JSONified list of device dictionaries
def getDevice(url):
    try:
        r = requests.get(url)
        return r.status_code, r.json()
    except requests.exceptions.JSONDecodeError:
        print('getDevice: error in get(', url, ')', sep='')
        print('exiting!')
        sys.exit()

# set IP Address for specified device
def setIPAddress(token, device, ipVal):
    global BASE_URL
    url = BASE_URL + 'set'
    pload = {'token':token,
             'hostname':device,
             'attribute': 'ipaddr',
             'value':ipVal}
    r = requests.post(url,data = pload)
    # print('STATUS CODE',r.status_code) # debug
    # print(r.text)
    return r.status_code, r.text

# handle failures: print diagnostics and exit (does not return!)
def handleFail(result):
    print('failed!')
    print('\tresponse code = ', result[0])
    print('\tresponse json = ', result[1])
    sys.exit()

def main():
    global USER, PASSWORD, BASE_URL

```

```

# result is a tuple containing response code and text (may be empty)
result = 0, ''

print("accessing FishNet server")

result = login(BASE_URL + 'login')
if result[0] != status.HTTP_200_OK:
    sys.exit()    # can't do anything if we fail to log in

print('\tcredentials OK')

# get our token from the response text
token = result[1].strip()
# print('\ttoken=', token) # debug

# get the device dictionary
print('getting full inventory')
result = getDevice(BASE_URL + 'get')
if result[0] != status.HTTP_200_OK:
    handleFail(result)

print(result[1]) # print the inventory

# get one device
print('getting switch1')
result = getDevice(BASE_URL + 'get?hostname=switch1')
if result[0] != status.HTTP_200_OK:
    handleFail(result)

print(result[1]) # print the device

# set switch1's IP address
print('setting switch1's ip address')
newIPAddr = '192.168.1.10'
result = setIPAddress(token, 'switch1', newIPAddr)
if result[0] != status.HTTP_200_OK:
    handleFail(result)

# print(result[1]) # debug

# get switch1's configuration to verify
print('getting switch1 again to verify')
result = getDevice(BASE_URL + 'get?hostname=switch1')
if result[0] != status.HTTP_200_OK:
    handleFail(result)

print('final result = ', result[1])
if result[1]['ipaddr'] == newIPAddr:
    print('addresses match!')
else:
    print('address do not match!')

if __name__ == "__main__":
    main()

```