Add device

# Metodo agregar dispositivo, manda configuración al NSO mediante API

def add\_device(nso\_socket, hostname, ipaddr, platform):

    logging.debug(f'!!! NSO\_ZTP\_SCRIPT: Started.')

    device\_sufix\_name, ned = get\_platform(platform)

    url = 'http://' + nso\_socket + '/restconf/data/tailf-ncs:devices'

    header = {

        'Authorization': 'Basic YWRtaW46YWRtaW4=',

        'Accept': 'application/yang-data+json',

        'Content-Type': 'application/yang-data+xml'

    }

    payload = (

        '<device>\n'

        '   <name>' + hostname + '</name>\n'

        '   <address>' + ipaddr + '</address>\n'

        '   <description>ZTP' + device\_sufix\_name + '</description>\n'

        '   <authgroup>ztp\_authgrp</authgroup>\n'

        '   <device-type>\n'

        '   <cli>\n' + ned + '\n'

        '   </cli>\n'

        '   </device-type>\n'

        '   <state>\n'

        '   <admin-state>unlocked</admin-state>\n'

        '   </state>\n'

        '</device>\n'

    )

    response = requests.post(url, headers=header, data=payload)

    logging.debug(f'!!! NSO\_ZTP\_SCRIPT: Device Added... RESTCONF Response Code: {response.status\_code}')

    return response.status\_code

Obtener llaves de acceso --- fetch-host-keys

# Metodo agregar llaves SSDH del dispositivo, manda configuración al NSO mediante API

def device\_ssh\_keys(nso\_socket, hostname):

    url = 'http://' + nso\_socket + '/restconf/data/tailf-ncs:devices/device=' + hostname + '/ssh/fetch-host-keys'

    header = {

        'Authorization': 'Basic YWRtaW46YWRtaW4=',

        'Accept': 'application/yang-data+json',

    }

    response = requests.post(url, headers=header)

    logging.debug(f'!!! NSO\_ZTP\_SCRIPT: SSH Keys Fetched... RESTCONF Response Code: {response.status\_code}')

    return response.status\_code

sincronizar dispositivo -- sync-from

# Metodo sincorniza el dispositivo, manda configuración al NSO mediante API

def device\_sync\_from(nso\_socket, hostname):

    url = 'http://' + nso\_socket + '/restconf/data/tailf-ncs:devices/device=' + hostname + '/sync-from'

    header = {

        'Authorization': 'Basic YWRtaW46YWRtaW4=',

        'Accept': 'application/yang-data+json',

    }

    response = requests.post(url, headers=header)

    logging.debug(f'!!! NSO\_ZTP\_SCRIPT: Device Synced...RESTCONF Response Code: {response.status\_code}')

    return response.status\_code

Carga golden config -- nso\_golden\_config:nso\_golden\_config

# Metodo manda golden config al NSO mediante API

def create\_golden\_config\_service(nso\_socket, service\_name, platform, hostname, ipaddr, region):

    url = 'http://' + nso\_socket + '/restconf/data/tailf-ncs:services/nso\_golden\_config:nso\_golden\_config/'

    header = {

        'Authorization': 'Basic YWRtaW46YWRtaW4=',

        'Accept': 'application/yang-data+json',

        'Content-Type': 'application/yang-data+xml'

    }

    payload = (

        '<nso\_golden\_config xmlns=\"http://example.com/nso\_golden\_config\">\n'

        '    <service\_instance>' + service\_name + '</service\_instance>\n'

        '    <platform>' + platform + '</platform>\n'

        '    <device>' + hostname + '</device>\n'

        '    <hostname>' + hostname + '</hostname>\n'

        '    <mgmt\_ip\_addr>' + ipaddr + '</mgmt\_ip\_addr>\n'

        '    <Region>' + region + '</Region>\n'

        '</nso\_golden\_config>'

    )

    response = requests.patch(url, headers=header, data=payload)

    logging.debug(f'!!! NSO\_ZTP\_SCRIPT: GOLDEN CONFIG SERVICE... RESTCONF Response Code: {response.status\_code}')

    return response.status\_code

