

Changing a Guest's desired state, waiting for result

This code, connects to Manager, authenticates and gets a list of all Guests present in that installation.

Implements the following arguments:

- **-m**: Manager's address (required)
- **-g**: Guest where this action is going to be applied (required)
- **-a**: Action to be executed on the Guest (required, either "start", "startonce", "stop", "suspend", "resume", "pause", "resume" or "shutdown")
- **-p**: Manager's password (optional, if not present, password will be asked)

```
#!/usr/bin/env python

import ssl
# Recent Python versions require a valid certificate. You can comment this
# if you've loaded a valid certificate in your Manager instance
if hasattr(ssl, '_create_unverified_context'):
    ssl._create_default_https_context = ssl._create_unverified_context

import sys
import json
import time
import getpass
from optparse import OptionParser
from optparse import OptionGroup
import urllib2
from urllib2 import Request, urlopen, URLError, HTTPError
from cookielib import CookieJar

if __name__ == "__main__":
    ACTIONS = ["start", "startonce", "stop", "suspend", "resume", "pause", "resume", "shutdown"]
    parser = OptionParser(conflict_handler="resolve")
    parser.add_option("-m", "--manager", dest="manager",
                    help="connect to this Manager ", metavar="MANAGER")
    parser.add_option("-p", "--password", dest="password",
                    help="admin password", metavar="PASSWORD")
    parser.add_option("-g", "--guest", dest="guestId",
                    help="guest to be started", metavar="GUESTID")
    parser.add_option("-a", "--action", dest="action",
                    help="action to be applied on guest", metavar="ACTION")

    if len(sys.argv) < 2:
        parser.print_help()
        sys.exit(-1)

    (options, args) = parser.parse_args(sys.argv[1:])

    if options.manager is None:
        parser.print_help()
        sys.exit(-1)

    if options.guestId is None:
        parser.print_help()
        sys.exit(-1)

    if options.action is None:
        parser.print_help()
        sys.exit(-1)
    elif not options.action in ACTIONS:
        print 'Option ' + options.action + ' is not supported.'
        print 'Try one of these: ' + str(ACTIONS)
        print
        sys.exit(-1)

    if options.password is None:
        options.password = getpass.getpass('Enter admin password: ')

    # We need to use CookieJar to store the session ID injected by Manager
    cj = CookieJar()
```

```

opener = urllib2.build_opener(urllib2.HTTPCookieProcessor(cj))

# Prepare an Authenticate Object
authenticate = {'username': 'admin',
               'password': options.password}

# Prepare urllib2 request, converting auth_object to JSON in the process
authUrl = 'https://' + options.manager + '/authenticate'
request = urllib2.Request(authUrl, json.dumps(authenticate))
request.add_header('Content-type', 'application/json')
response = opener.open(request)
rawData = response.read()

# Try to parse response as an AuthResult Object
try:
    authResult = json.loads(rawData)
except Exception as ex:
    print 'Can\'t parse Manager response as AuthResult Object'
    print 'Data: ' + rawData
    print
    sys.exit(-1)
if 'status' in authResult:
    if authResult['status'] != 'OK':
        print 'Manager reject this credentials'
        print
        sys.exit(-1)
else:
    print 'Invalid answer from Manager'
    print 'JSON Data: ' + authResult
    print
    sys.exit(-1)

# Prepare a GuestAction object
guestAction = {'repeat': '0',
              'retries': '0',
              'type': 'guest_action',
              'guest_id': options.guestId,
              'action': options.action}
createTaskUrl = 'https://' + options.manager + '/tasks/create'
request = urllib2.Request(createTaskUrl, json.dumps(guestAction))
request.add_header('Content-type', 'application/json')
response = opener.open(request)
rawData = response.read()

# Try to parse answer as a TaskResult Object
try:
    taskResult = json.loads(rawData)
    status = taskResult['status']
    taskId = taskResult['message']
except Exception as ex:
    print 'Can\'t parse Manager answer a TaskResult Object'
    print 'Data: ' + rawData
    print
    sys.exit(-1)

if status != "OK":
    print 'There was an error trying to create Task'
    print 'Error: ' + taskId
    print
    sys.exit(-1)

print 'Task created with ID=' + taskId

# We have to use sys.stdout.write instead print because:
# 1.- We don't want a newline
# 2.- Python sucks
sys.stdout.write('Waiting for completion...')
sys.stdout.flush()

count = 0
# Wait for task completion to obtain its result

```

```

while True:
    # Sleep a little to give some time to complete the Task
    time.sleep(1)

    # Request the state for this Task
    thisTaskUrl = 'https://' + options.manager + '/tasks/' + taskId
    request = urllib2.Request(thisTaskUrl)
    response = opener.open(request)
    rawData = response.read()

    # Try to parse answer as a JSON dictionary
    try:
        taskState = json.loads(rawData)
    except Exception as ex:
        print 'Can\'t parse Manager answer as a JSON dictionary'
        print 'Data: ' + rawData
        print
        sys.exit(-1)

    # Break the Task has completed
    if taskState['state'] == 'completed':
        break

    # Bail out if we've waiting for more than 10 seconds
    if count > 10:
        print 'Timed out waiting for Task completion'
        print
        sys.exit(-1)

    count += 1
    sys.stdout.write('.')
    sys.stdout.flush()

print ' done'

# Get task result and notify user
result = taskState['result']
message = taskState['result_msg']
if result != 'success':
    print 'Action ' + options.action + ' failed on guest ' + options.guestId
    print 'Message from Manager: ' + message
    print
    sys.exit(-1)
else:
    print 'Action ' + options.action + ' succeed on guest ' + options.guestId
    print 'Message from Manager: ' + message
    sys.exit(0)

```