

```

import zlib
from optparse import OptionParser
import httplib
import sys

#proxy_server_ip_port = "113.20.129.50:8080"
proxy_server_ip_port = "api.s.mel.secureinf.net"
auth_uri = "/auth/v0"
log_account = "log_data_account"
log_user = "log_data_user"
log_pass = "log_data_pass"
log_container = "log_processing_data"
processing_data_container_uri = "/v0/"+log_account+"/"+log_container+"/"

def unzip_file(filename):

    d = zlib.decompressobj(16 + zlib.MAX_WBITS)

    dataFile = open(filename)

    chunk = dataFile.read()

    chunk = d.decompress(chunk)

    print chunk

    return 0

def unzip_string(string):

    d = zlib.decompressobj(16 + zlib.MAX_WBITS)

    return d.decompress(string)

def get_file_list(token):

    #curl -H 'X-Auth-Token: AUTH_tke26da28c25fd4ca78e2e18d847b1f390' http://
    113.20.129.50:8080/v1/AUTH_log_data_account/log_processing_data
    conn = httplib.HTTPSConnection(proxy_server_ip_port)

    headers = {"X-Auth-Token": token}

    conn.request("GET", processing_data_container_uri, "", headers)

    res = conn.getresponse()

    if res.status != 200:
        print ("get file list error: %d" % res.status)
        return -1

```

```

return res.read()

def get_token():

    #curl -i -H "X-Auth-User: AUTH_testa27:AUTH_testu27" -H "X-Auth-Key: testp" http://
113.20.129.50:8080/auth/v1.0
    conn = httplib.HTTPSConnection(proxy_server_ip_port)

    headers = {"X-Auth-User": log_account+":"+log_user, "X-Auth-Key": log_pass}

    conn.request("GET", auth_uri, "", headers)

    res = conn.getresponse()

    if res.status != 200:
        print ("get token error: %d" % res.status)
        return -1

    return res.getheader("X-Auth-Token")

def get_file_name(options, data):

    #2011/12/09/09/30fb4508d42d9b256e58b491f59eedb6.csv.gz
    for line in data.split("\n"):

        if line == "processed_files.pickle.gz":
            continue

        arr = line.split('/')

        if arr[0] is "":
            continue

        if int(arr[0]) == options.year:
            if int(arr[1]) == options.month:
                if int(arr[2]) == options.day:
                    if int(arr[3]) == options.hour:
                        return line

    return -1

def download_one_file(token, filename):

    conn = httplib.HTTPSConnection(proxy_server_ip_port)

    headers = {"X-Auth-Token": token}

    conn.request("GET", processing_data_container_uri+filename, "", headers)

```

```

res = conn.getresponse()

if res.status != 200:
    print ("download one file error: %d" % res.status)
    return -1

return unzip_string(res.read())

def list_all(options):

    token = get_token()

    if token is -1:
        print ("list all in one hour fault")
        return -1

    print "get token back: \n"+token

    data = get_file_list(token)

    if data is -1:
        print ("list all in one hour fault")
        return -2

    print "get data back: \n"+data

    filename = get_file_name(options, data)

    if filename is -1:
        print ("list all in one hour fault")
        return -3

    print "get file name: \n"+filename

    ret = download_one_file(token, filename)

    if ret is -1:
        print ("list all in one hour fault")
        return -4

    if options.account is None:
        print "\nlines:"
        print ret
    else:
        arr = ret.split('\n')
        account = options.account
        hour = "%02d" % (options.hour - 2)
        day = "%02d" % options.day

```

```

kv_k = []
kv_v = []
flag = 0

for key in arr[0].split(','):
    kv_k.append(key)

for i in range(len(arr)):

    if i is 0:
        continue

    #2011/12/08 14:00:00,
    if account == arr[i].split(',')[1] and hour == arr[i].split(',')[0].split(' ')[1].split(':')[0] and day ==
arr[i].split(',')[0].split(' ')[0].split('/')[2]:
        #if account == arr[i].split(',')[1]:
            print "\nline:"
            print arr[i]
            flag = 1

            for value in arr[i].split(','):
                kv_v.append(value)

if flag == 0:
    print "no data of this account in this hour"
    sys.exit(1)

print "\nanalysis:"

body = ""

for i in range(len(kv_k)):
    #print kv_k[i]+":"+kv_v[i]
    body += kv_k[i]+"="+kv_v[i)+"\n"

if options.url is not None:

    headers = {"X-Auth-User": account, "Content-length": len(body), "Content-
Type": "application/x-www-form-urlencoded"}
    domain, uri = options.url.split("/", 1)

    print domain
    print uri

    conn = httplib.HTTPConnection(domain)

    conn.request("POST", "/" + uri, body, headers)

    res = conn.getresponse()

```

```

        print body
        print res.status

    return 0

if __name__ == '__main__':
    parser = OptionParser(usage='Usage: %prog [options]')

    parser.add_option('-Y', type='int', dest='year',
        help='which year log file you want to download?')
    parser.add_option('-M', type='int', dest='month',
        help='which month log file you want to download?')
    parser.add_option('-D', type='int', dest='day',
        help='which day log file you want to download?')
    parser.add_option('-H', type='int', dest='hour',
        help='which hour log file you want to download?')
    parser.add_option('-A', type='string', dest='account',
        help='which account log record you want to show?')
    parser.add_option('-F', type='string', dest='filename',
        help='which file log you want to show?')
    parser.add_option('-U', type='string', dest='url',
        help='which url you want to post?')

    options, args = parser.parse_args()

    if options.filename is not None:
        unzip_file(options.filename)
        sys.exit(1)

    if options.year is None:
        print ("Error: missing -Y argument")
        sys.exit(1)

    if options.month is None:
        print ("Error: missing -M argument")
        sys.exit(1)

    if options.day is None:
        print ("Error: missing -D argument")
        sys.exit(1)

    if options.hour is None:
        print ("Error: missing -H argument")
        sys.exit(1)

    list_all(options)
    sys.exit(1)

```