'''

CloudService XBMC Plugin

Copyright (C) 2013-2014 ddurdle

This program is free software: you can redistribute it and/or modify

it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

GNU General Public License for more details.

You should have received a copy of the GNU General Public License

along with this program. If not, see <http://www.gnu.org/licenses/>.

'''

# cloudservice - required python modules

import sys

import urllib

import re

import os

# cloudservice - standard XBMC modules

import xbmc, xbmcgui, xbmcplugin, xbmcaddon, xbmcvfs

# common routines

from resources.lib import kodi\_common

# global variables

import addon\_parameters

addon = addon\_parameters.addon

cloudservice2 = addon\_parameters.cloudservice2

cloudservice1 = addon\_parameters.cloudservice1

#\*\*\* testing - gdrive

from resources.lib import tvWindow

from resources.lib import gSpreadsheets

from resources.lib import gSheets\_api4

##\*\*

# cloudservice - standard modules

#from resources.lib import gdrive

#from resources.lib import gdrive\_api2

from resources.lib import cloudservice

from resources.lib import authorization

from resources.lib import folder

from resources.lib import file

from resources.lib import offlinefile

from resources.lib import package

from resources.lib import mediaurl

from resources.lib import crashreport

from resources.lib import gPlayer

from resources.lib import settings

from resources.lib import cache

from resources.lib import TMDB

#global variables

PLUGIN\_URL = sys.argv[0]

plugin\_handle = None

plugin\_queries = None

try:

plugin\_handle = int(sys.argv[1])

plugin\_queries = settings.parse\_query(sys.argv[2][1:])

except: pass

addon\_dir = xbmc.translatePath( addon.getAddonInfo('path') )

kodi\_common.debugger()

# cloudservice - create settings module

settings = settings.settings(addon)

# retrieve settings

user\_agent = settings.getSetting('user\_agent')

#obsolete, replace, revents audio from streaming

#if user\_agent == 'Mozilla/4.0 (compatible; MSIE 5.5; Windows NT)':

# addon.setSetting('user\_agent', 'Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/532.0 (KHTML, like Gecko) Chrome/3.0.195.38 Safari/532.0')

instanceName = addon\_parameters.PLUGIN\_NAME + str(settings.getSetting('account\_default', 1))

service = cloudservice2(PLUGIN\_URL,addon,instanceName, user\_agent, settings)

# must load after all other (becomes blocking)

# streamer

if service is not None and service.settings.streamer:

from BaseHTTPServer import BaseHTTPRequestHandler,HTTPServer

from resources.lib import streamer

import urllib, urllib2

from SocketServer import ThreadingMixIn

import threading

try:

server = streamer.MyHTTPServer(('', service.settings.streamPort), streamer.myStreamer)

server.setAccount(service, '')

print "ENABLED STREAMER \n\n\n"

while server.ready:

server.handle\_request()

server.socket.close()

except: pass