'''

Copyright (C) 2014-2016 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/>.

'''

import urllib

import re

#

#

#

class file:

# CloudService v0.2.5

#

# - add resolution (video) [2015/06/20]

# - add playcount (video/music) [2015/06/20]

# - add checksum [2016/03/03]

# - add commands [2016/03/03]

AUDIO = 1

VIDEO = 2

PICTURE = 3

UNKNOWN = 4

##

##

def \_\_init\_\_(self, id, title, plot, type, fanart,thumbnail, date='', size=0, resolution=None, playcount=0, duration=-1, download='', checksum=''):

self.id = id

self.title = title

self.showtitle = title

self.plot = plot

self.type = type

self.fanart = fanart

self.thumbnail = thumbnail

self.download = download

self.hasMeta = False

self.isEncoded = False

self.date = date

self.size = size

self.srtURL = ''

self.resume = 0

self.decryptedTitle = ''

self.resolution = resolution

self.playcount = playcount

self.duration = duration

self.commands = ''

self.checksum = checksum

self.rating = ''

self.cast = ''

self.isTV = False

self.isMovie = False

self.cloudResume = 0

self.deleted = False

self.director = ''

self.set = ''

self.genre = ''

self.country = ''

self.year = ''

self.actors = None

self.isTV = False

self.TVID = None

self.MOVIEID = None

# nekwebdev contribution

self.regtv1 = re.compile('(.+?)'

'[ .]?[ \-]?\s\*S(\d\d?)E(\d\d?)'

'(.\*)'

'(?:[ .](\d{3}\d?p)|\Z)?'

'\..\*', re.IGNORECASE)

self.regtv2 = re.compile('(.+?)'

'[ .]?[ \-]?\s\*season\s?(\d\d?)\s?episode\s?(\d\d?)'

'(.\*)'

'(?:[ .](\d{3}\d?p)|\Z)?'

'\..\*', re.IGNORECASE)

self.regtv3 = re.compile('(.+?)'

'[ .]?[ \-]?\s\*(\d\d?)x(\d\d?)'

'(.\*)'

'(?:[ .](\d{3}\d?p)|\Z)?'

'\..\*', re.IGNORECASE)

self.regmovie = re.compile('(.\*?[ \(]?[ .]?[ \-]?\d{4}[ \)]?[ .]?[ \-]?)'

'.\*?'

'(?:(\d{3}\d?p)|\Z)?')

def setAlbumMeta(self,album,artist,releaseDate,trackNumber,genre, trackTitle):

self.album = album

self.artist = artist

self.trackNumber = trackNumber

self.genre = genre

self.releaseDate = releaseDate

self.trackTitle = trackTitle

self.hasMeta = True

def setTVMeta(self,show,season,episode,showtitle):

self.show = show

self.season = season

self.episode = episode

self.showtitle = showtitle

self.hasMeta = True

self.isTV = True

def displayTitle(self):

if self.decryptedTitle != '':

return urllib.unquote(str(self.decryptedTitle) + ' [' + str(self.title) + ']')

else:

return urllib.unquote(self.title)

def displayShowTitle(self):

if self.decryptedTitle != '':

return urllib.unquote(str(self.decryptedTitle) + ' [' + str(self.title) + ']')

elif self.showtitle is not None and self.showtitle != '':

return urllib.unquote(self.showtitle)

else:

return urllib.unquote(self.title)

def displayTrackTitle(self):

if self.decryptedTitle != '':

return urllib.unquote(str(self.decryptedTitle) + ' [' + str(self.title) + ']')

elif self.showtitle is not None and self.trackTitle != '':

return urllib.unquote(self.trackTitle)

else:

return urllib.unquote(self.title)

def \_\_repr\_\_(self):

return '{}: {} {}'.format(self.\_\_class\_\_.\_\_name\_\_,

self.title)

def \_\_cmp\_\_(self, other):

if hasattr(other, 'title'):

return self.title.\_\_cmp\_\_(other.title)

def getKey(self):

return self.title