Final Project

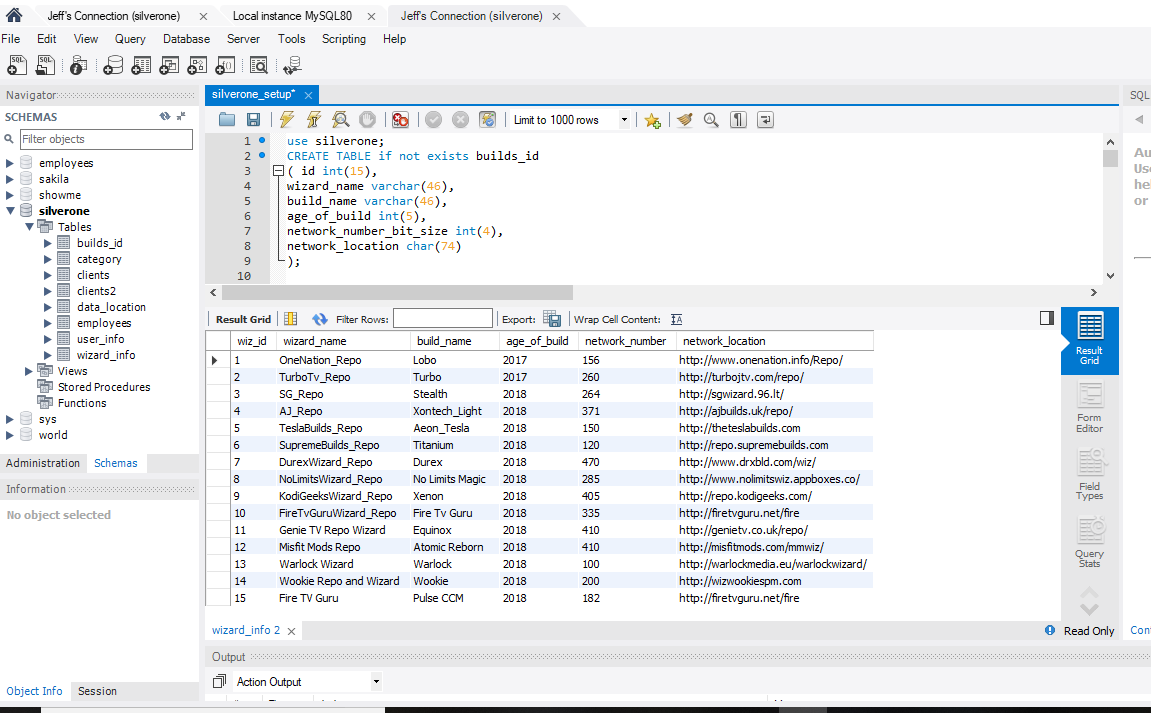
Jeff Fitih

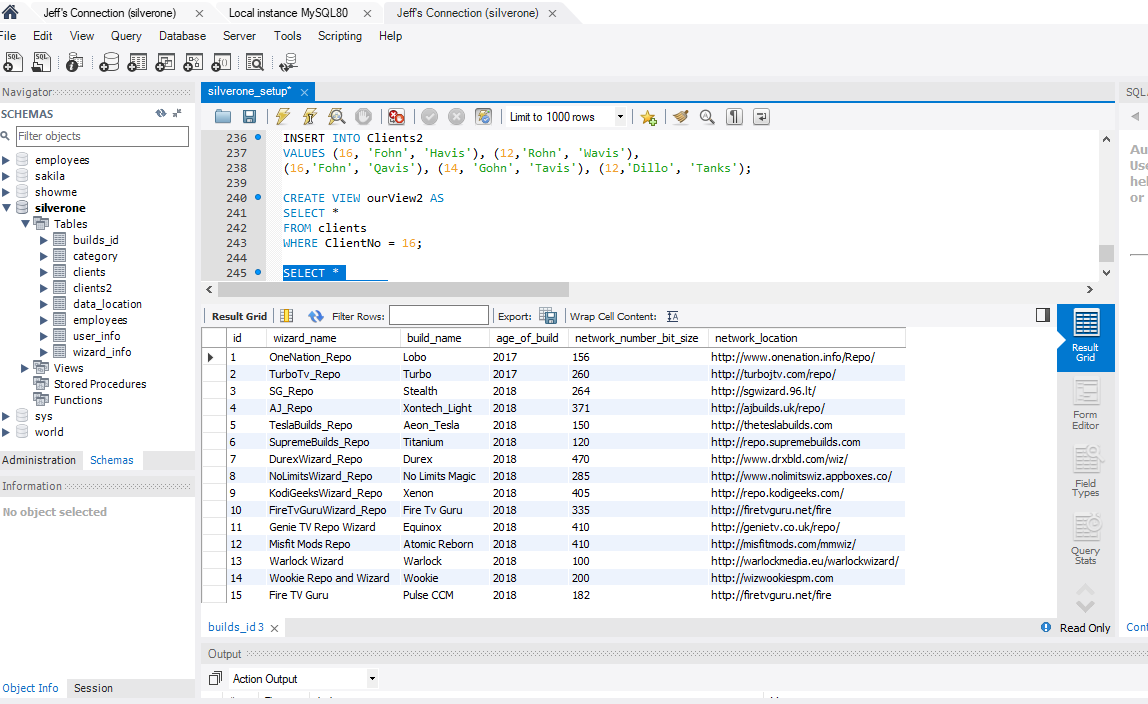
For my final project I would like to have Kodi repos to populate my database with repos to work in my database. To have the data to be retrieved from a Google Drive to run as a server to my URL.

My python code was from the github forum. <forum>https://github.com/cguZZman/plugin.googledrive/issues</forum>

Which was provided so I can make my own repo for others to bounce off of. I can say I am trying to make a stop shop for all your entertainment needs. Fire has a lot of streaming. But the problem I am have is finding the correct format. Because Apple has a different format than Google and Google has a different format than Android. I want once a person hits my URL they grab my zip drive that will have all add-ons without having my wizard being compromised. I will keep populating SQL with more builds and add-ons.

* Open-source software originally designed for Microsoft Xbox
* Can be installed on any media- TV, desktop, laptop, smartphone, and tablet
* Compatible on all platforms, including Mac, Android, Windows, iOS, and Linux
* Used to watch free movies, live streaming of music shows and sporting events, international TV shows, and more
* Download Kodi installation that works for your device
* Download Fusion Installer
* Run configuration wizard, or Addon Installer
* Config Wizard is automated, whereas Addon Installer allows you to choose Kodi Addons before installing them
* Takes no more than 15 minutes to install
* Access to thousands of live TV channels
* Unlimited access to international TV shows and live events
* Easy to use
* Safe
* Addons are like apps
* Popular KodiAddons include 1Channel, cCloud- TV, and USTVnow Plus
* Addons are installed under different sections
* Categories include pictures, videos, music, etc.
* Videos can be seen only from a specific year and earlier: This could be because you
* haven’t set the proper date and time on the Kodi.
* No stream available: This has nothing to do with Kodi Addon. You’ll have to search for another stream
* TV ADDONS offers a platform for several third party developers
* I want to offer thousands of Kodi Addons for users to choose from





use silverone;

CREATE TABLE if not exists builds\_id

( id int(15),

wizard\_name varchar(46),

build\_name varchar(46),

age\_of\_build int(5),

network\_number\_bit\_size int(4),

network\_location char(74)

);

INSERT INTO builds\_id(id, wizard\_name, build\_name, )

VALUES (01, 'OneNation\_Repo', 'Lobo', '2017', '156', 'http://www.onenation.info/Repo/'),

(02,'TurboTv\_Repo', 'Turbo', '2017', 260, 'http://turbojtv.com/repo/'),

(03,'SG\_Repo', 'Stealth', '2018', 264, 'http://sgwizard.96.lt/'),

(04, 'AJ\_Repo', 'Xontech\_Light', '2018', 371, 'http://ajbuilds.uk/repo/'),

(05,'TeslaBuilds\_Repo', 'Aeon\_Tesla', '2018', 150, 'http://theteslabuilds.com'),

(06,'SupremeBuilds\_Repo', 'Titanium', '2018', 120, 'http://repo.supremebuilds.com'),

(07,'DurexWizard\_Repo', 'Durex', '2018', 470, 'http://www.drxbld.com/wiz/'),

(08,'NoLimitsWizard\_Repo', 'No Limits Magic', '2018', 285, 'http://www.nolimitswiz.appboxes.co/'),

(09,'KodiGeeksWizard\_Repo', 'Xenon', '2018', 405, 'http://repo.kodigeeks.com/'),

(10,'FireTvGuruWizard\_Repo', 'Fire Tv Guru', '2018', 335, 'http://firetvguru.net/fire'),

(11,'Genie TV Repo Wizard', 'Equinox', '2018', 410, 'http://genietv.co.uk/repo/'),

(12,'Misfit Mods Repo', 'Atomic Reborn', '2018', 410, 'http://misfitmods.com/mmwiz/'),

(13,'Warlock Wizard', 'Warlock', '2018', 100, 'http://warlockmedia.eu/warlockwizard/'),

(14,'Wookie Repo and Wizard', 'Wookie', '2018', 200, 'http://wizwookiespm.com'),

(15,'Fire TV Guru', 'Pulse CCM', '2018', 182, 'http://firetvguru.net/fire');

CREATE TABLE if not exists employees

( id int(5),

name\_f varchar(46),

dept varchar(46),

age int(2),

salary int(10),

location varchar(46)

);

INSERT INTO builds\_id

VALUES (01, 'OneNation\_Repo', 'Lobo', '2017', '156', 'http://www.onenation.info/Repo/'),

(02,'TurboTv\_Repo', 'Turbo', '2017', 260, 'http://turbojtv.com/repo/'),

(03,'SG\_Repo', 'Stealth', '2018', 264, 'http://sgwizard.96.lt/'),

(04, 'AJ\_Repo', 'Xontech\_Light', '2018', 371, 'http://ajbuilds.uk/repo/'),

(05,'TeslaBuilds\_Repo', 'Aeon\_Tesla', '2018', 150, 'http://theteslabuilds.com'),

(06,'SupremeBuilds\_Repo', 'Titanium', '2018', 120, 'http://repo.supremebuilds.com'),

(07,'DurexWizard\_Repo', 'Durex', '2018', 470, 'http://www.drxbld.com/wiz/'),

(08,'NoLimitsWizard\_Repo', 'No Limits Magic', '2018', 285, 'http://www.nolimitswiz.appboxes.co/'),

(09,'KodiGeeksWizard\_Repo', 'Xenon', '2018', 405, 'http://repo.kodigeeks.com/'),

(10,'FireTvGuruWizard\_Repo', 'Fire Tv Guru', '2018', 335, 'http://firetvguru.net/fire'),

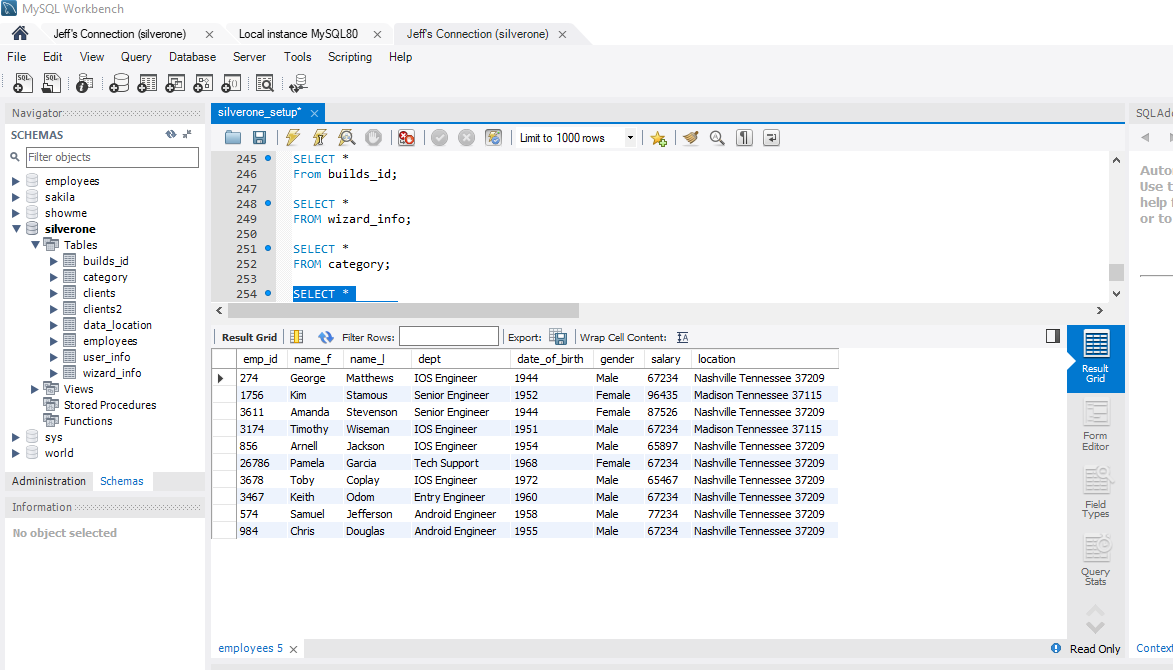
(11,'Genie TV Repo Wizard', 'Equinox', '2018', 410, 'http://genietv.co.uk/repo/'),

(12,'Misfit Mods Repo', 'Atomic Reborn', '2018', 410, 'http://misfitmods.com/mmwiz/'),

(13,'Warlock Wizard', 'Warlock', '2018', 100, 'http://warlockmedia.eu/warlockwizard/'),

(14,'Wookie Repo and Wizard', 'Wookie', '2018', 200, 'http://wizwookiespm.com'),

(15,'Fire TV Guru', 'Pulse CCM', '2018', 182, 'http://firetvguru.net/fire');



CREATE TABLE if not exists category

( id int(15),

wizard\_name varchar(46),

build\_name varchar(46),

age\_of\_build int(5),

network\_number int(46),

network\_location char(46)

);

INSERT INTO builds\_id

VALUES (01, 'OneNation\_Repo', 'Lobo', '2017', '156', 'http://www.onenation.info/Repo/'),

(02,'TurboTv\_Repo', 'Turbo', '2017', 260, 'http://turbojtv.com/repo/'),

(03,'SG\_Repo', 'Stealth', '2018', 264, 'http://sgwizard.96.lt/'),

(04, 'AJ\_Repo', 'Xontech\_Light', '2018', 371, 'http://ajbuilds.uk/repo/'),

(05,'TeslaBuilds\_Repo', 'Aeon\_Tesla', '2018', 150, 'http://theteslabuilds.com'),

(06,'SupremeBuilds\_Repo', 'Titanium', '2018', 120, 'http://repo.supremebuilds.com'),

(07,'DurexWizard\_Repo', 'Durex', '2018', 470, 'http://www.drxbld.com/wiz/'),

(08,'NoLimitsWizard\_Repo', 'No Limits Magic', '2018', 285, 'http://www.nolimitswiz.appboxes.co/'),

(09,'KodiGeeksWizard\_Repo', 'Xenon', '2018', 405, 'http://repo.kodigeeks.com/'),

(10,'FireTvGuruWizard\_Repo', 'Fire Tv Guru', '2018', 335, 'http://firetvguru.net/fire'),

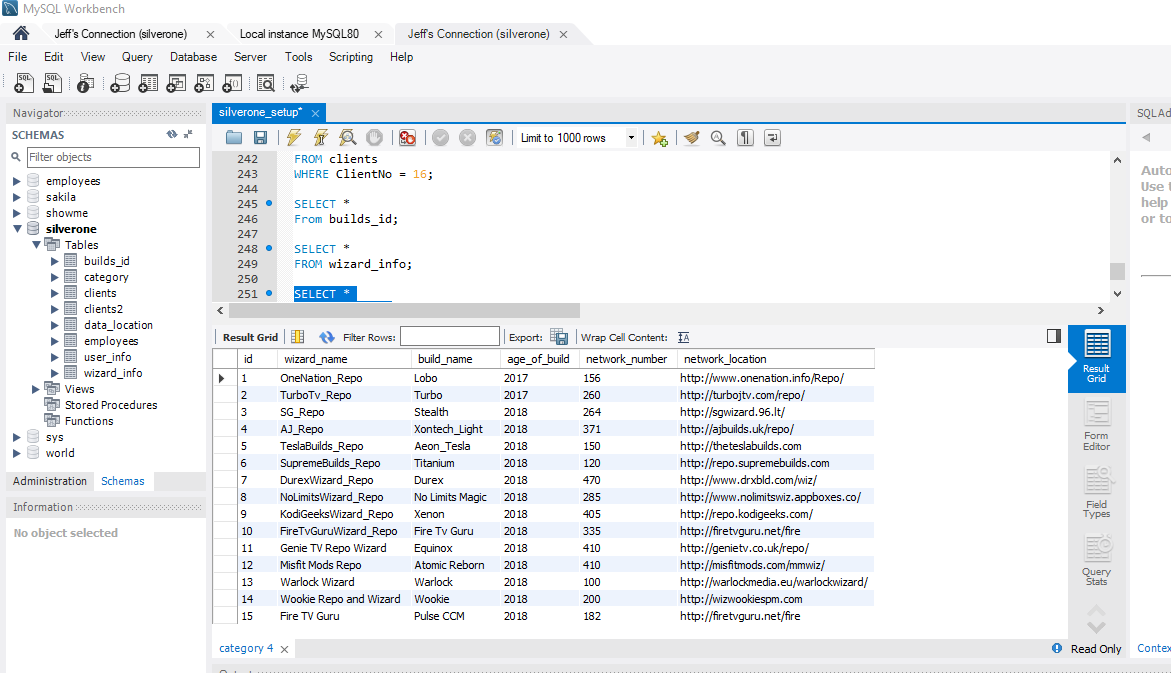
(11,'Genie TV Repo Wizard', 'Equinox', '2018', 410, 'http://genietv.co.uk/repo/'),

(12,'Misfit Mods Repo', 'Atomic Reborn', '2018', 410, 'http://misfitmods.com/mmwiz/'),

(13,'Warlock Wizard', 'Warlock', '2018', 100, 'http://warlockmedia.eu/warlockwizard/'),

(14,'Wookie Repo and Wizard', 'Wookie', '2018', 200, 'http://wizwookiespm.com'),

(15,'Fire TV Guru', 'Pulse CCM', '2018', 182, 'http://firetvguru.net/fire');



CREATE TABLE if not exists data\_location

( data\_id int(5),

name\_f varchar(46),

dept varchar(46),

age int(2),

salary int(10),

location varchar(46)

);

INSERT INTO builds\_id

VALUES (01, 'OneNation\_Repo', 'Lobo', '2017', '156', 'http://www.onenation.info/Repo/'),

(02,'TurboTv\_Repo', 'Turbo', '2017', 260, 'http://turbojtv.com/repo/'),

(03,'SG\_Repo', 'Stealth', '2018', 264, 'http://sgwizard.96.lt/'),

(04, 'AJ\_Repo', 'Xontech\_Light', '2018', 371, 'http://ajbuilds.uk/repo/'),

(05,'TeslaBuilds\_Repo', 'Aeon\_Tesla', '2018', 150, 'http://theteslabuilds.com'),

(06,'SupremeBuilds\_Repo', 'Titanium', '2018', 120, 'http://repo.supremebuilds.com'),

(07,'DurexWizard\_Repo', 'Durex', '2018', 470, 'http://www.drxbld.com/wiz/'),

(08,'NoLimitsWizard\_Repo', 'No Limits Magic', '2018', 285, 'http://www.nolimitswiz.appboxes.co/'),

(09,'KodiGeeksWizard\_Repo', 'Xenon', '2018', 405, 'http://repo.kodigeeks.com/'),

(10,'FireTvGuruWizard\_Repo', 'Fire Tv Guru', '2018', 335, 'http://firetvguru.net/fire'),

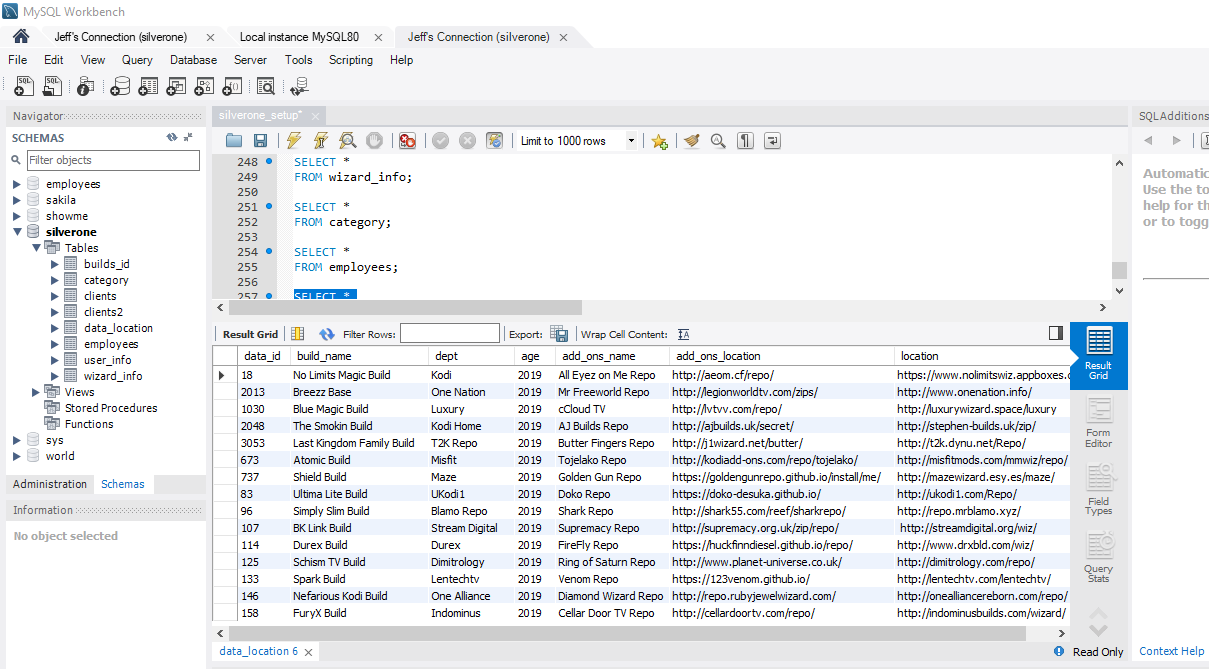
(11,'Genie TV Repo Wizard', 'Equinox', '2018', 410, 'http://genietv.co.uk/repo/'),

(12,'Misfit Mods Repo', 'Atomic Reborn', '2018', 410, 'http://misfitmods.com/mmwiz/'),

(13,'Warlock Wizard', 'Warlock', '2018', 100, 'http://warlockmedia.eu/warlockwizard/'),

(14,'Wookie Repo and Wizard', 'Wookie', '2018', 200, 'http://wizwookiespm.com'),

(15,'Fire TV Guru', 'Pulse CCM', '2018', 182, 'http://firetvguru.net/fire');



CREATE TABLE if not exists wizard\_info

( wiz\_id int(15),

wizard\_name varchar(46),

build\_name varchar(46),

age\_of\_build int(5),

network\_number int(46),

network\_location varchar(46)

);

INSERT INTO builds\_id

VALUES (01, 'OneNation\_Repo', 'Lobo', '2017', '156', 'http://www.onenation.info/Repo/'),

(02,'TurboTv\_Repo', 'Turbo', '2017', 260, 'http://turbojtv.com/repo/'),

(03,'SG\_Repo', 'Stealth', '2018', 264, 'http://sgwizard.96.lt/'),

(04, 'AJ\_Repo', 'Xontech\_Light', '2018', 371, 'http://ajbuilds.uk/repo/'),

(05,'TeslaBuilds\_Repo', 'Aeon\_Tesla', '2018', 150, 'http://theteslabuilds.com'),

(06,'SupremeBuilds\_Repo', 'Titanium', '2018', 120, 'http://repo.supremebuilds.com'),

(07,'DurexWizard\_Repo', 'Durex', '2018', 470, 'http://www.drxbld.com/wiz/'),

(08,'NoLimitsWizard\_Repo', 'No Limits Magic', '2018', 285, 'http://www.nolimitswiz.appboxes.co/'),

(09,'KodiGeeksWizard\_Repo', 'Xenon', '2018', 405, 'http://repo.kodigeeks.com/'),

(10,'FireTvGuruWizard\_Repo', 'Fire Tv Guru', '2018', 335, 'http://firetvguru.net/fire'),

(11,'Genie TV Repo Wizard', 'Equinox', '2018', 410, 'http://genietv.co.uk/repo/'),

(12,'Misfit Mods Repo', 'Atomic Reborn', '2018', 410, 'http://misfitmods.com/mmwiz/'),

(13,'Warlock Wizard', 'Warlock', '2018', 100, 'http://warlockmedia.eu/warlockwizard/'),

(14,'Wookie Repo and Wizard', 'Wookie', '2018', 200, 'http://wizwookiespm.com'),

(15,'Fire TV Guru', 'Pulse CCM', '2018', 182, 'http://firetvguru.net/fire');

CREATE TABLE if not exists user\_info

( user\_id int(5),

name\_f varchar(46),

name\_l varchar(46),

username varchar(46),

email varchar(255),

age int(2),

signal\_strength int(10),

location varchar(46)

);

INSERT INTO builds\_id

VALUES (01, 'OneNation\_Repo', 'Lobo', '2017', '156', 'http://www.onenation.info/Repo/'),

(02,'TurboTv\_Repo', 'Turbo', '2017', 260, 'http://turbojtv.com/repo/'),

(03,'SG\_Repo', 'Stealth', '2018', 264, 'http://sgwizard.96.lt/'),

(04, 'AJ\_Repo', 'Xontech\_Light', '2018', 371, 'http://ajbuilds.uk/repo/'),

(05,'TeslaBuilds\_Repo', 'Aeon\_Tesla', '2018', 150, 'http://theteslabuilds.com'),

(06,'SupremeBuilds\_Repo', 'Titanium', '2018', 120, 'http://repo.supremebuilds.com'),

(07,'DurexWizard\_Repo', 'Durex', '2018', 470, 'http://www.drxbld.com/wiz/'),

(08,'NoLimitsWizard\_Repo', 'No Limits Magic', '2018', 285, 'http://www.nolimitswiz.appboxes.co/'),

(09,'KodiGeeksWizard\_Repo', 'Xenon', '2018', 405, 'http://repo.kodigeeks.com/'),

(10,'FireTvGuruWizard\_Repo', 'Fire Tv Guru', '2018', 335, 'http://firetvguru.net/fire'),

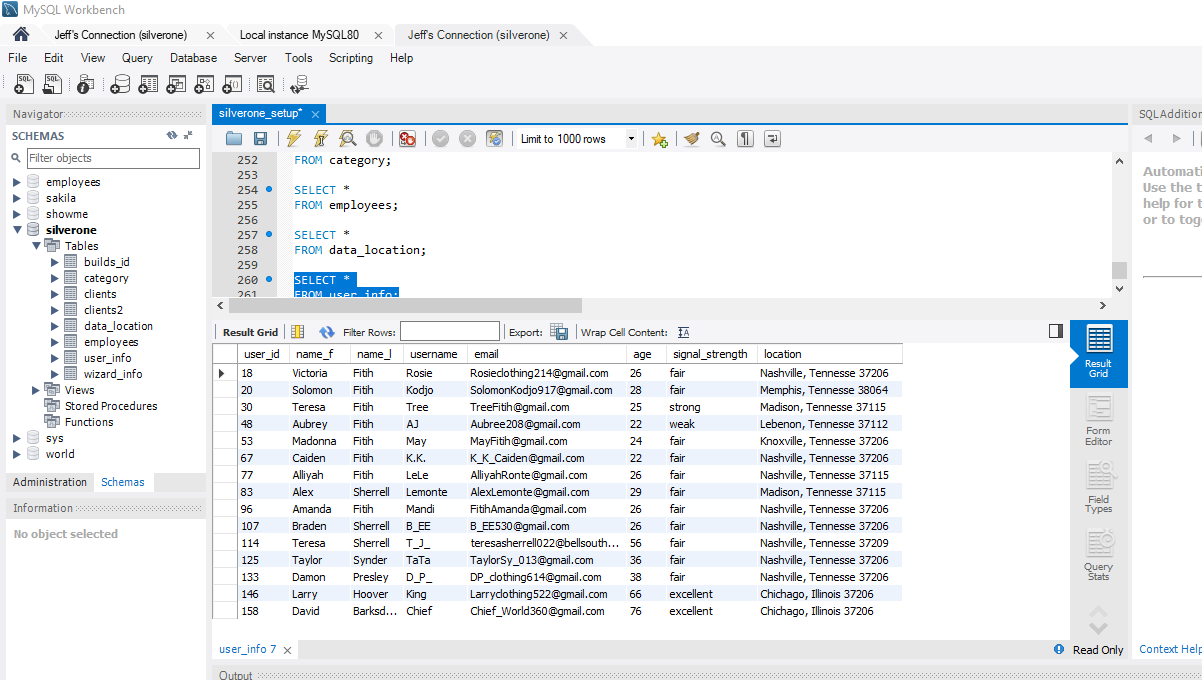
(11,'Genie TV Repo Wizard', 'Equinox', '2018', 410, 'http://genietv.co.uk/repo/'),

(12,'Misfit Mods Repo', 'Atomic Reborn', '2018', 410, 'http://misfitmods.com/mmwiz/'),

(13,'Warlock Wizard', 'Warlock', '2018', 100, 'http://warlockmedia.eu/warlockwizard/'),

(14,'Wookie Repo and Wizard', 'Wookie', '2018', 200, 'http://wizwookiespm.com'),

(15,'Fire TV Guru', 'Pulse CCM', '2018', 182, 'http://firetvguru.net/fire');



CREATE TABLE IF NOT EXISTS Clients (

ClientNo INT NOT NULL,

First\_Name VARCHAR (45) NULL,

Last\_Name VARCHAR (45) NULL

);

INSERT INTO Clients(ClientNo, First\_Name, Last\_Name)

VALUES (16, 'Fohn', 'Havis'), (12,'Rohn', 'Wavis'),

(16,'Fohn', 'Qavis'), (14, 'Gohn', 'Tavis'), (12,'Dillo', 'Tanks');

CREATE VIEW ourView2 AS

SELECT \*

FROM clients

WHERE ClientNo = 16;

CREATE VIEW ourView AS

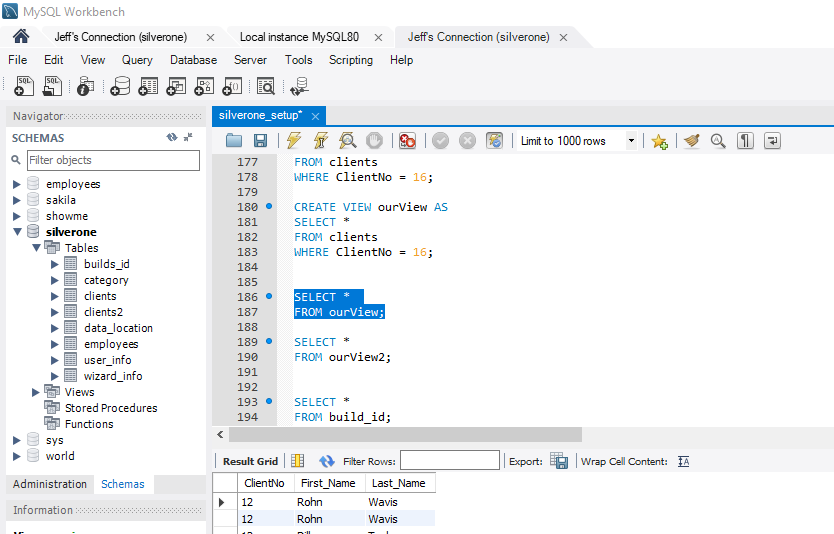
SELECT \*

FROM clients

WHERE ClientNo = 16;

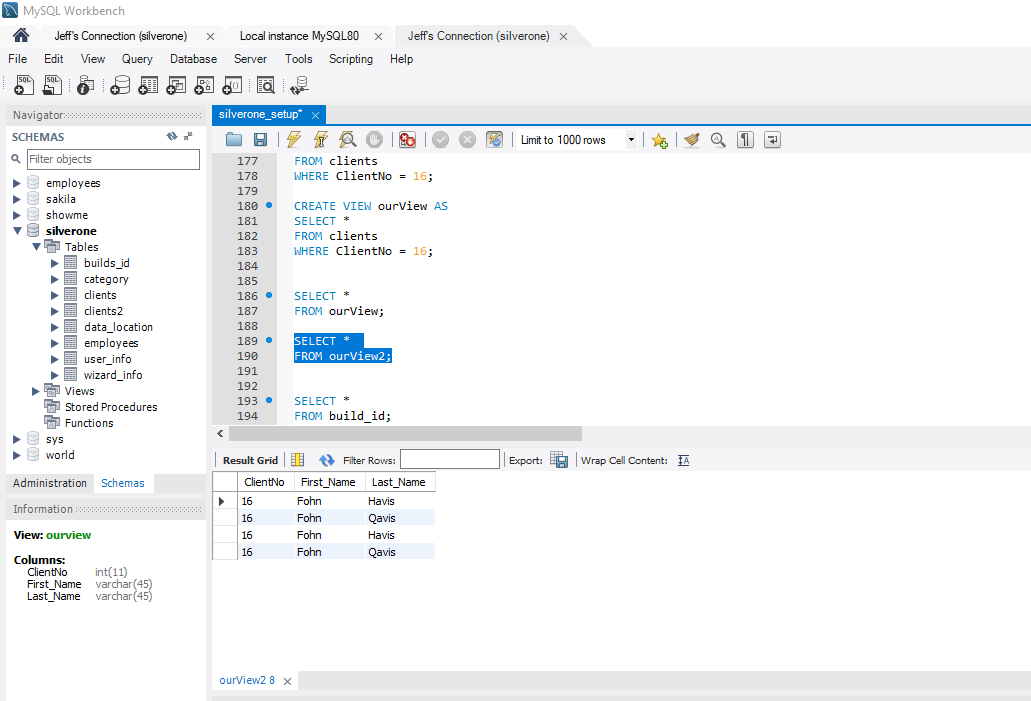
SELECT \*

FROM ourView;



SELECT \*

FROM ourView2;



SELECT \*

FROM build\_id;

SET SQL\_SAFE\_UPDATES = 0;

INSERT INTO Clients VALUES (NULL, 'Fillo', 'Uanks');

UPDATE Clients

SET ClientNo = 15

WHERE ClientNo = 12;

UPDATE Clients

SET ClientNo = 12

WHERE ClientNo = NULL ;

DELETE FROM Clients

WHERE ClientNo = 12;

DROP ourView

DROP Clients;

CREATE TABLE IF NOT EXISTS Clients2 (

ClientNo INT NOT NULL,

First\_Name VARCHAR (45) NULL,

Last\_Name VARCHAR (45) NULL

);

CREATE VIEW ourView2 AS

SELECT \*

FROM clients

WHERE ClientNo = 16;

INSERT INTO Clients2

VALUES (16, 'Fohn', 'Havis'), (12,'Rohn', 'Wavis'),

(16,'Fohn', 'Qavis'), (14, 'Gohn', 'Tavis'), (12,'Dillo', 'Tanks');

CREATE VIEW ourView2 AS

SELECT \*

FROM clients

WHERE ClientNo = 16;

INSERT INTO Clients2

VALUES (16, 'Fohn', 'Havis'), (12,'Rohn', 'Wavis'),

(16,'Fohn', 'Qavis'), (14, 'Gohn', 'Tavis'), (12,'Dillo', 'Tanks');

CREATE VIEW ourView2 AS

SELECT \*

FROM clients

WHERE ClientNo = 16;

SELECT \*

From builds\_id;

SELECT \*

FROM wizard\_info;

# Google Drive KODI Addon

Google Drive for KODI

Play all your media from Google Drive including Videos, Music and Pictures.

\* Unlimited accounts

\* Playback your music and videos. Listing of videos with thumbnails.

\* Use Google Drive as a source.

\* Subtitles can be assigned automatically if a .str file exists with the same name as the video.

\* Export your videos to your library (.strm files). You can export your music too, but kodi won't support it yet. It's a Kodi issue for now.

\* Show your photos individually or run a slideshow of them. Listing of pictures with thumbnails.

\* Auto-Refreshed slideshow.

\* Use of OAuth 2 login. You don't have to write your user/password within the add-on. Use the login process in your browser.

\* Extremely fast. Using the Google Drive API

This program is not affiliated with or sponsored by Google.

252  addon.py

@@ -0,0 +1,252 @@

#-------------------------------------------------------------------------------

# Copyright (C) 2017 Carlos Guzman (cguZZman) carlosguzmang@protonmail.com

#

# This file is part of Google Drive for Kodi

#

# Google Drive for Kodi 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.

#

# Cloud Drive Common Module for Kodi 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 datetime

import urllib

from clouddrive.common.cache.simplecache import SimpleCache

from clouddrive.common.ui.addon import CloudDriveAddon

from clouddrive.common.utils import Utils

from resources.lib.provider.googledrive import GoogleDrive

from clouddrive.common.ui.logger import Logger

from urllib2 import HTTPError

class GoogleDriveAddon(CloudDriveAddon):

\_provider = GoogleDrive()

\_parameters = {'spaces': 'drive', 'prettyPrint': 'false'}

\_file\_fileds = 'id,name,mimeType,description,hasThumbnail,thumbnailLink,modifiedTime,owners(permissionId),parents,size,imageMediaMetadata(width),videoMediaMetadata'

\_cache = None

\_child\_count\_supported = False

\_change\_token = None

def \_\_init\_\_(self):

self.\_cache = SimpleCache()

super(GoogleDriveAddon, self).\_\_init\_\_()

def get\_provider(self):

return self.\_provider

def get\_my\_files\_menu\_name(self):

return self.\_addon.getLocalizedString(32013)

def get\_custom\_drive\_folders(self, driveid):

self.\_account\_manager.load()

drive\_folders = []

drive\_folders.append({'name' : self.\_common\_addon.getLocalizedString(32058), 'path' : 'sharedWithMe'})

if self.\_content\_type == 'image':

drive\_folders.append({'name' : self.\_addon.getLocalizedString(32007), 'path' : 'photos'})

drive\_folders.append({'name' : self.\_addon.getLocalizedString(32014), 'path' : 'starred'})

return drive\_folders

def new\_change\_token\_slideshow(self, change\_token, driveid, item\_driveid=None, item\_id=None, path=None):

self.\_provider.configure(self.\_account\_manager, driveid)

if not change\_token:

response = self.\_provider.get('/changes/startPageToken', parameters = self.\_parameters)

self.\_change\_token = Utils.get\_safe\_value(response, 'startPageToken')

change\_token = 1

else:

Logger.notice('checking change with %s and %s' % (self.\_change\_token, change\_token))

page\_token = self.\_change\_token

while page\_token:

self.\_parameters['pageToken'] = page\_token

self.\_parameters['fields'] = 'nextPageToken,newStartPageToken,changes(file(id,name,parents))'

Logger.notice('%s' % (self.\_parameters))

response = self.\_provider.get('/changes', parameters = self.\_parameters)

Logger.notice(response)

if self.cancel\_operation():

return

self.\_change\_token = Utils.get\_safe\_value(response, 'newStartPageToken', self.\_change\_token)

changes = Utils.get\_safe\_value(response, 'changes', [])

for change in changes:

f = Utils.get\_safe\_value(change, 'file', {})

Logger.notice(f)

parents = Utils.get\_safe\_value(f, 'parents', [])

parents.append(f['id'])

if item\_id in parents:

return change\_token + 1

page\_token = Utils.get\_safe\_value(response, 'nextPageToken')

return change\_token

def get\_folder\_items(self, driveid, item\_driveid=None, item\_id=None, path=None, on\_items\_page\_completed=None):

self.\_provider.configure(self.\_account\_manager, driveid)

item\_driveid = Utils.default(item\_driveid, driveid)

self.\_parameters['fields'] = 'files(%s),nextPageToken' % self.\_file\_fileds

if item\_id:

self.\_parameters['q'] = '\'%s\' in parents' % item\_id

elif path == 'sharedWithMe' or path == 'starred':

self.\_parameters['q'] = path

elif path == 'photos':

self.\_parameters['spaces'] = path

else:

if path == '/':

self.\_parameters['q'] = '\'root\' in parents'

else:

item = self.get\_item\_by\_path(path)

self.\_parameters['q'] = '\'%s\' in parents' % item['id']

files = self.\_provider.get('/files', parameters = self.\_parameters)

if self.cancel\_operation():

return

return self.process\_files(driveid, files, on\_items\_page\_completed)

def search(self, query, driveid, item\_driveid=None, item\_id=None, on\_items\_page\_completed=None):

self.\_provider.configure(self.\_account\_manager, driveid)

item\_driveid = Utils.default(item\_driveid, driveid)

self.\_parameters['fields'] = 'files(%s)' % self.\_file\_fileds

query = 'fullText contains \'%s\'' % query

if item\_id:

query += ' and \'%s\' in parents' % item\_id

self.\_parameters['q'] = query

files = self.\_provider.get('/files', parameters = self.\_parameters)

if self.cancel\_operation():

return

return self.process\_files(driveid, files, on\_items\_page\_completed)

def process\_files(self, driveid, files, on\_items\_page\_completed=None):

items = []

for f in files['files']:

item = self.\_extract\_item(f)

cache\_key = self.\_addonid+'-drive-'+driveid+'-item\_driveid-'+item['drive\_id']+'-item\_id-'+item['id']+'-path-None'

self.\_cache.set(cache\_key, f, expiration=datetime.timedelta(minutes=1))

items.append(item)

if on\_items\_page\_completed:

on\_items\_page\_completed(items)

if 'nextPageToken' in files:

self.\_parameters['pageToken'] = files['nextPageToken']

next\_files = self.\_provider.get('/files', parameters = self.\_parameters)

if self.cancel\_operation():

return

items.extend(self.process\_files(driveid, next\_files, on\_items\_page\_completed))

return items

def \_extract\_item(self, f, include\_download\_info=False):

size = long('%s' % Utils.get\_safe\_value(f, 'size', 0))

item = {

'id': f['id'],

'name': f['name'],

'name\_extension' : Utils.get\_extension(f['name']),

'drive\_id' : Utils.get\_safe\_value(Utils.get\_safe\_value(f, 'owners', [{}])[0], 'permissionId'),

'mimetype' : Utils.get\_safe\_value(f, 'mimeType'),

'last\_modified\_date' : Utils.get\_safe\_value(f,'modifiedTime'),

'size': size,

'description': Utils.get\_safe\_value(f, 'description', '')

}

if item['mimetype'] == 'application/vnd.google-apps.folder':

item['folder'] = {

'child\_count' : 0

}

if 'videoMediaMetadata' in f:

video = f['videoMediaMetadata']

item['video'] = {

'width' : Utils.get\_safe\_value(video, 'width'),

'height' : Utils.get\_safe\_value(video, 'height'),

'duration' : long('%s' % Utils.get\_safe\_value(video, 'durationMillis', 0)) / 1000

}

if 'audio' in f:

audio = f['audio']

item['audio'] = {

'tracknumber' : Utils.get\_safe\_value(audio, 'track'),

'discnumber' : Utils.get\_safe\_value(audio, 'disc'),

'duration' : int(Utils.get\_safe\_value(audio, 'duration') or '0') / 1000,

'year' : Utils.get\_safe\_value(audio, 'year'),

'genre' : Utils.get\_safe\_value(audio, 'genre'),

'album': Utils.get\_safe\_value(audio, 'album'),

'artist': Utils.get\_safe\_value(audio, 'artist'),

'title': Utils.get\_safe\_value(audio, 'title')

}

if 'imageMediaMetadata' in f:

item['image'] = {

'size' : size

}

if 'hasThumbnail' in f and f['hasThumbnail']:

item['thumbnail'] = Utils.get\_safe\_value(f, 'thumbnailLink')

if include\_download\_info:

item['download\_info'] = {

'url' : self.\_provider.\_get\_api\_url() + '/files/%s?alt=media&access\_token=%s' % (f['id'], self.\_provider.get\_access\_tokens()['access\_token'])

}

return item

def get\_item\_by\_path(self, path, include\_download\_info=False):

if path[:1] == '/':

path = path[1:]

if path[-1:] == '/':

path = path[:-1]

Logger.notice('get\_item\_by\_path: %s' % path)

parts = path.split('/')

parent = 'root'

current\_path = ''

item = None

self.\_parameters['fields'] = 'files(%s)' % self.\_file\_fileds

for part in parts:

part = urllib.unquote(part)

current\_path += '/%s' % part

self.\_parameters['q'] = '\'%s\' in parents and name = \'%s\'' % (parent, part)

files = self.\_provider.get('/files', parameters = self.\_parameters)

if (len(files['files']) > 0):

for f in files['files']:

item = self.\_extract\_item(f, include\_download\_info)

parent = item['id']

cache\_key = self.\_addonid+'-drive-None-item\_driveid-None-item\_id-None-path-'+current\_path

self.\_cache.set(cache\_key, f, expiration=datetime.timedelta(minutes=1))

break

else:

item = None

break

if not item:

raise HTTPError(path, 404, 'Not found', None, None)

return item

def get\_item(self, driveid, item\_driveid=None, item\_id=None, path=None, find\_subtitles=False, include\_download\_info=False):

self.\_provider.configure(self.\_account\_manager, driveid)

item\_driveid = Utils.default(item\_driveid, driveid)

cache\_key = self.\_addonid+'-drive-'+driveid+'-item\_driveid-'+Utils.str(item\_driveid)+'-item\_id-'+Utils.str(item\_id)+'-path-'+Utils.str(path)

f = self.\_cache.get(cache\_key)

if f:

item = self.\_extract\_item(f, include\_download\_info)

else:

cache\_key = self.\_addonid+'-drive-None-item\_driveid-None-item\_id-None-path-'+path

f = self.\_cache.get(cache\_key)

if f:

item = self.\_extract\_item(f, include\_download\_info)

else:

self.\_parameters['fields'] = self.\_file\_fileds

if not item\_id and path == '/':

item\_id = 'root'

if item\_id:

f = self.\_provider.get('/files/%s' % item\_id, parameters = self.\_parameters)

self.\_cache.set(cache\_key, f, expiration=datetime.timedelta(seconds=59))

item = self.\_extract\_item(f, include\_download\_info)

else:

item = self.get\_item\_by\_path(path, include\_download\_info)

if find\_subtitles:

subtitles = []

self.\_parameters['q'] = 'name contains \'%s\'' % urllib.quote(Utils.remove\_extension(item['name']))

files = self.\_provider.get('/files', parameters = self.\_parameters)

for f in files['files']:

subtitle = self.\_extract\_item(f, include\_download\_info)

if subtitle['name\_extension'] == 'srt' or subtitle['name\_extension'] == 'sub' or subtitle['name\_extension'] == 'sbv':

subtitles.append(subtitle)

if subtitles:

item['subtitles'] = subtitles

return item

if \_\_name\_\_ == '\_\_main\_\_':

GoogleDriveAddon().route()

38  addon.xml

@@ -0,0 +1,38 @@

<?xml version="1.0" encoding="UTF-8"?>

<addon id="plugin.googledrive" name="Google Drive" version="1.0.0" provider-name="Carlos Guzman (cguZZman)">

<requires>

<import addon="xbmc.python" version="2.25.0" />

<import addon="script.module.clouddrive.common"/>

</requires>

<extension point="xbmc.python.pluginsource" library="addon.py">

<provides>image audio video</provides>

</extension>

<extension point="xbmc.service" library="service.py" start="login" />

<extension point="xbmc.addon.metadata">

<platform>all</platform>

<summary lang="en\_GB">Google Drive for KODI</summary>

<description lang="en\_GB">

Play all your media from Google Drive including Videos, Music and Pictures.

- Unlimited number of personal or business accounts.

- Search over your drive.

- Auto-Refreshed slideshow.

- Export your videos to your library (.strm files)

- Use Google Drive as a source

- This program is not affiliated with or sponsored by Google.

</description>

<summary lang="he\_IL">כונן Google Drive של מיקרוסופט עבור קודי</summary>

<description lang="he\_IL">

הפעל את כל המדיה שלך מ- Google Drive כולל וידאו, מוסיקה ותמונות.

  - מספר בלתי מוגבל של חשבונות אישיים או עסקיים.

  - חיפוש ומציאת הכונן שלך.

  - ריענון מצגת אוטומטית.

  - ייצוא קטעי הווידאו לספרייה שלך (קבצי .strm)

  - שימוש ב-Google Drive כמקור

  - תוכנית זו אינה מזוהה עם או בחסות מיקרוסופט.

</description>

<license>GNU GENERAL PUBLIC LICENSE. Version 3, 29 June 2007</license>

<source>https://github.com/cguZZman/plugin.googledrive</source>

<forum>https://github.com/cguZZman/plugin.googledrive/issues</forum>

<website>https://addons.kodi.tv/show/plugin.googledrive</website>

</extension>

</addon>