

Mount Your Remote Google Music Library Locally on Linux with GMusicFS

MTE maketecheasier.com/mount-remote-google-music-library-locally

Google Music users: Have you ever wished you could mount your entire library from Google's servers in a folder on your Linux desktop? If so, GMusicFS may interest you. It's a python tool that works with Fuse to make it possible to mount Google Music files locally. Here's how to get it going.

Note: GMusicFS is under development and may give you issues. In some cases, it may not even work. Use at your own risk.

Installing dependencies

Before GMusicFS can function, some of its needs must be met. The first of which is a package known as Pip. It allows for the installation of python packages. Install it in your terminal by entering the following command:

```
sudo apt-get install python-pip
```

Now that the tool needed has been installed, it's time to install GMusicFS's dependencies. Fortunately, only two need to be installed. The first of which is the unofficial Google Music API. Without this, there would be no way to interact with Google Music on any level.

```
pip install https://github.com/simon-weber/Unofficial-Google-Music-API/tarball/d
develop
Collecting https://github.com/simon-weber/Unofficial-Google-Music-API/tarball/de
velop
  Downloading https://github.com/simon-weber/Unofficial-Google-Music-API/tarball
/develop (147kB)
    100% |#####| 147kB 1.3MB/s
```

```
pip install https://github.com/simon-weber/Unofficial-Google-Music-API/tarball/develop
```

Another dependency that must be installed is known as Fuse. This tool allows interaction with file systems. Install it with Pip.

```
pip install https://github.com/terencehonles/fusepy/tarball/master
Collecting https://github.com/terencehonles/fusepy/tarball/master
  Downloading https://github.com/terencehonles/fusepy/tarball/master
```

```
pip install https://github.com/terencehonles/fusepy/tarball/master
```

Installing GMusicFS

```
usage: gmusicfs [-h] [-f] [-v] [-vv] [-t] [--allusers] [--nolibrary]
               [--deviceid]
               mountpoint

GMusicFS

positional arguments:
  mountpoint            The location to mount to

optional arguments:
  -h, --help            show this help message and exit
  -f, --foreground      Don't daemonize, run in the foreground.
  -v, --verbose         Be a little verbose
  -vv, --veryverbose    Be very verbose
  -t, --truefilesize    Report true file sizes (slower directory reads)
  --allusers            Allow all system users access to files (Requires
                        user_allow_other set in /etc/fuse.conf)
  --nolibrary           Don't scan the library at launch
  --deviceid            Get the device ids bounded to your account
```

Much like the two dependencies this program requires, GMusicFS can be easily installed with the Pip program. Just enter this one command, and it should be installed onto your system.

```
pip install https://github.com/EnigmaCurry/GMusicFS/tarball/master
```

Configuring GMusicFS

Just installing GMusicFS with python isn't enough. A configuration file must be created in order for the program to work. Using the terminal, enter the following:

```
nano ~/.gmusicfs
```

This will launch the nano text editor and create a file in your home directory at the same time. Inside nano, paste the following:

```
[credentials]
username = your_username@gmail.com
password = your_password
deviceId = your_mobile_id
```

```
GNU nano 2.4.2      File: /home/derrik/.gmusicfs

[credentials]
username = 
password = 
deviceId = 

[ Read 5 lines ]

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

After pasting the text above into nano, you'll need to change a few things. In the "username" section put the Gmail account you use Google Music with. In "password" enter your password. Save nano (with keyboard shortcut "Ctrl + o"), but don't exit. Some more information will need to be opened later. You'll need to generate a device ID to use.

Open a second terminal window. Inside that terminal window enter `gmusicfs --deviceid`. It'll generate an ID for you to use. When you have your generated ID, paste it into the nano configuration area and save the file again.

Note: after saving `~/gmusicfs`, it's a good idea to run `chmod 600 ~/gmusicfs`.

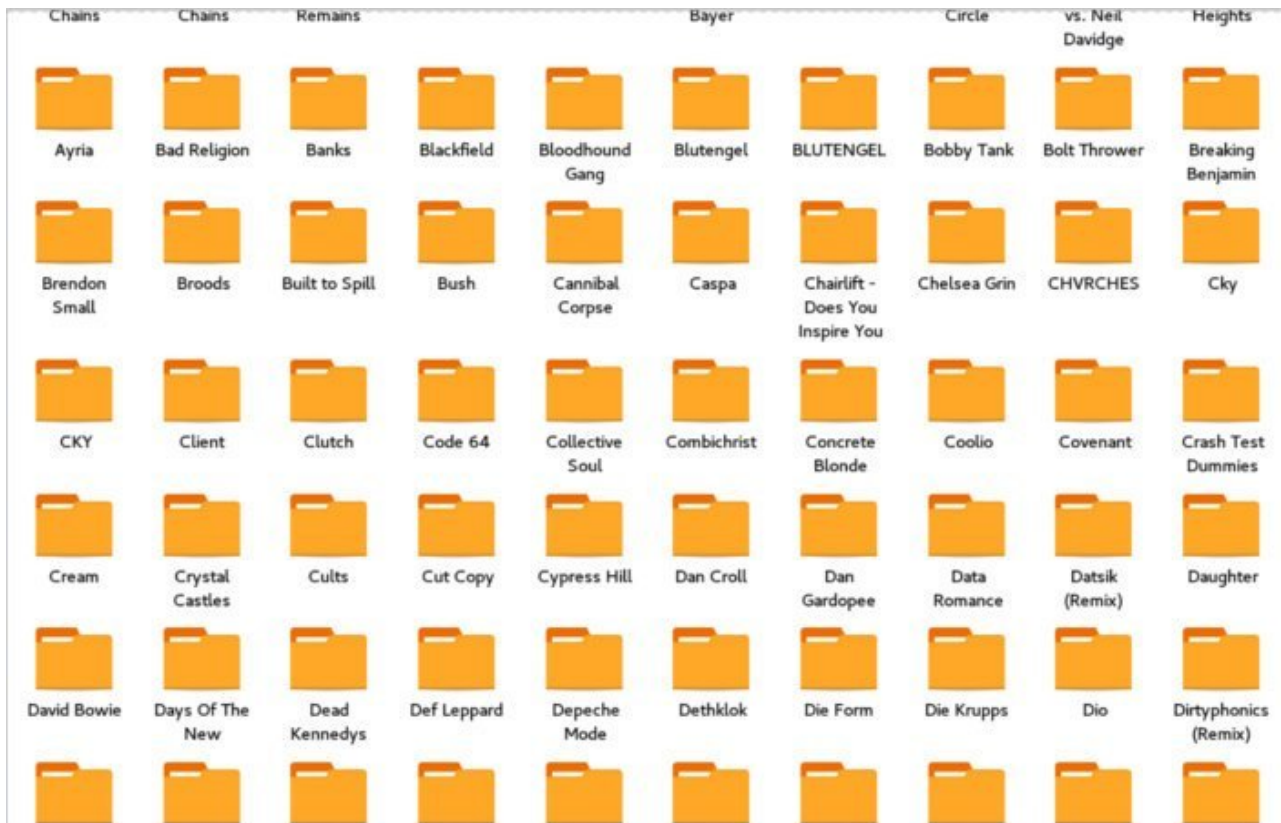
Using GMusicFS

Mounting a Google music library with GMusicFS is fairly simple. First, create a directory for it to be mounted to.

```
mkdir -p ~/google_music_files
```

After that just run the mount command:

```
gmusicfs ~/google_music_files
```



GMusicFS can also be unmounted:

```
fusermount -u ~/google_music_files
```

With Google Music mounted to the local file system, it's incredibly easy to use this mount point to add files to your favorite music player. Just point the player at “~/google_music_files” (or wherever else you decide to mount) and go!

Conclusion

GMusicFS is great. It allows you to combine two awesome things: Google's cloud music storage and the power of local file access. With this tool users won't be tethered down to just listening to their uploaded collection within the designated Google app. Instead, users can enjoy their music in any application they choose.