MellowPlayer Documentation

Release 3.7.0

Colin Duquesnoy

Sep 18, 2021

Contents

MellowPlayer

Cloud music integration for your desktop !

CHAPTER 1

User documentation

Contents:

1.1 About MellowPlayer

MellowPlayer is born from the need of a Qt-based alternative to NuvolaPlayer for the KaOS linux distribution

Here are the initial goals:

- the application should embed a web view of the music streaming service (the same as you see in a regular browser) and should provide an integration with the desktop (media keys support, global shortcuts, notifications,. . .).
- the application should be able to support more than 1 streaming service
- we (the core team) will only support the streaming services that we are actively using. Other services should be added and maintained by contributors. The main reason is that we won't be able to support non-free services (even those who have a trial period). Support for free services (even with limitations) might be added by the team after the release 1.0.
- adding a new service/extension should be easy: you just write a javascript plugin

1.2 Installation

This page will guide you throught the installation of MellowPlayer on the supported operating systems.

1.2.1 GNU/Linux

We provide several ways to install a pre-compiled version of MellowPlayer on GNU/Linux:

1. Flatpak

2. Native packages

Flatpak is the recommended solution as it is what the developers use and test against. It contains a recent version of QtWebEngine/Chromium which is essential for MellowPlayer to work correctly.

Native packages (especially on old/LTS distributions) may cause problems as they don't provide a recent version of QtWebEngine/Chromium

Flatpak

MellowPlayer is available on flathub:

```
flatpak install flathub com.gitlab.ColinDuquesnoy.MellowPlayer
```

Native Packages

Fedora

Starting from Fedora 27, MellowPlayer is available from the official stable repositories:

```
sudo dnf install mellowplayer
```

Most services require proprietary audio codecs to work. You can install them from the RPMFusion repositories:

```
sudo dnf install qt5-qtwebengine-freeworld
```

ArchLinux

MellowPlayer is available from the AUR, install it with your favorite AUR tool (e.g. yaourt).

```
yaourt -S mellowplayer
```

KaOS

MellowPlayer is available from KaOSx/apps repository, just run:

```
$ sudo pacman -S mellowplayer
```

Other distributions

Pre-compiled packages for other distributions (Ubuntu, openSUSE,. . .) can be found on our OBS Download Page

Compiling from source

See the README for build instructions.

Widevine DRM Plugin

Many services like Spotify, Tidal and Amazon Music require the widevine DRM plugin to work.

You can install it on GNU/Linux by running the below script (tested with native packages and flatpak; make sure the *binutils* package is installed on Debian/Ubuntu providing the *ar* command)

```
curl -s "https://gitlab.com/ColinDuquesnoy/MellowPlayer/-/raw/master/scripts/install-
↪widevine.sh" | bash
```

1.2.2 Windows

Just grab the windows installer from the official website (click on the **Windows folder**) and follow the instructions.

Please note the Windows Installer we provide is built with a version of QtWebEngine built without proprietary codecs support (for licensing reasons). If your favorite service require proprietary codecs to work, you'll need to build QtWebEngine with the flag `use_proprietary_codecs` and build MellowPlayer using that QtWebEngine version.

1.2.3 OS X

OSX is not officially supported anymore. You may try to build and run MellowPlayer from sources.

1.3 Features

- Cross-platform (available on Windows, Mac OSX and GNU/Linux)
- System tray integration and notifications
- Mpris2 support (GNU/Linux only)
- Hotkeys and media player keys support
- Plugin based application (you can add support for a new web-based music streaming service by writing a **javascript plugin**)
- User scripts support

1.3.1 User Scripts

With MellowPlayer >= 3.1 you have the possibility to use user scripts.

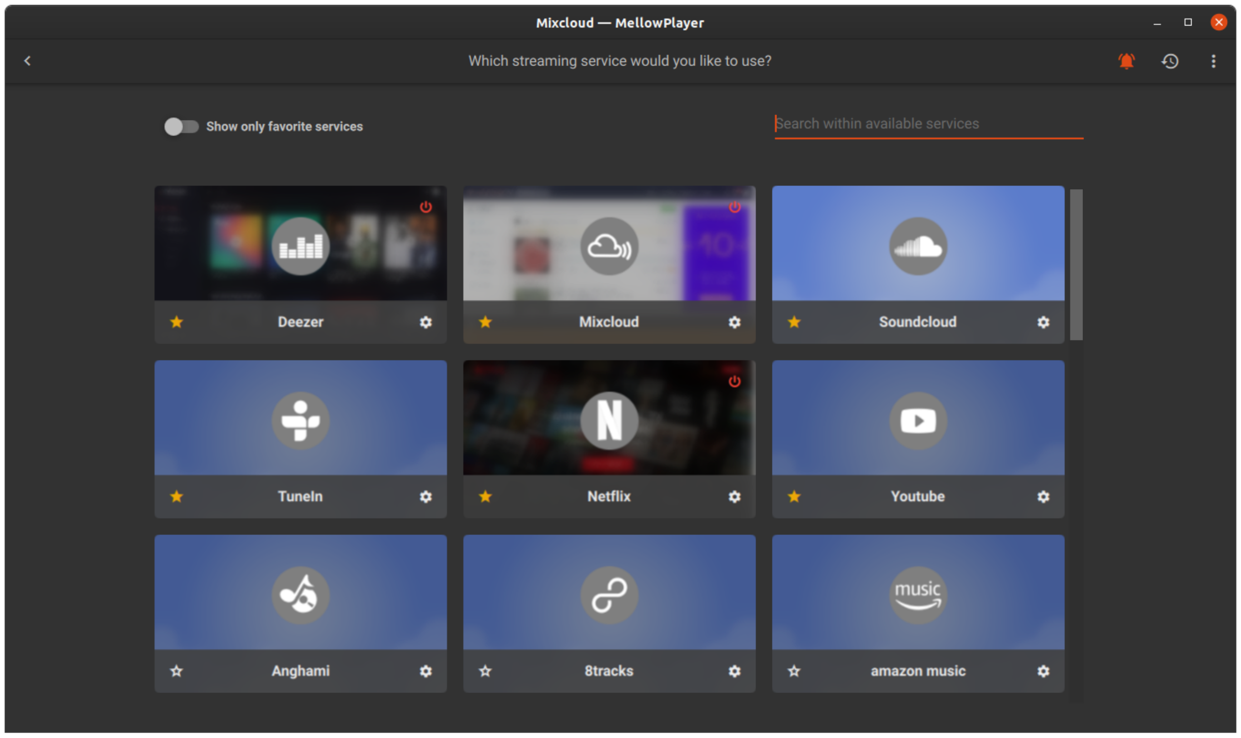
This means that you can customize the look and feel of a streaming service as you like it, or simply to add features you miss.

- Download and use different themes
- Download and use different user scripts

1.4 Getting started

1.4.1 First startup

On the first startup, you'll be presented with the following screen:

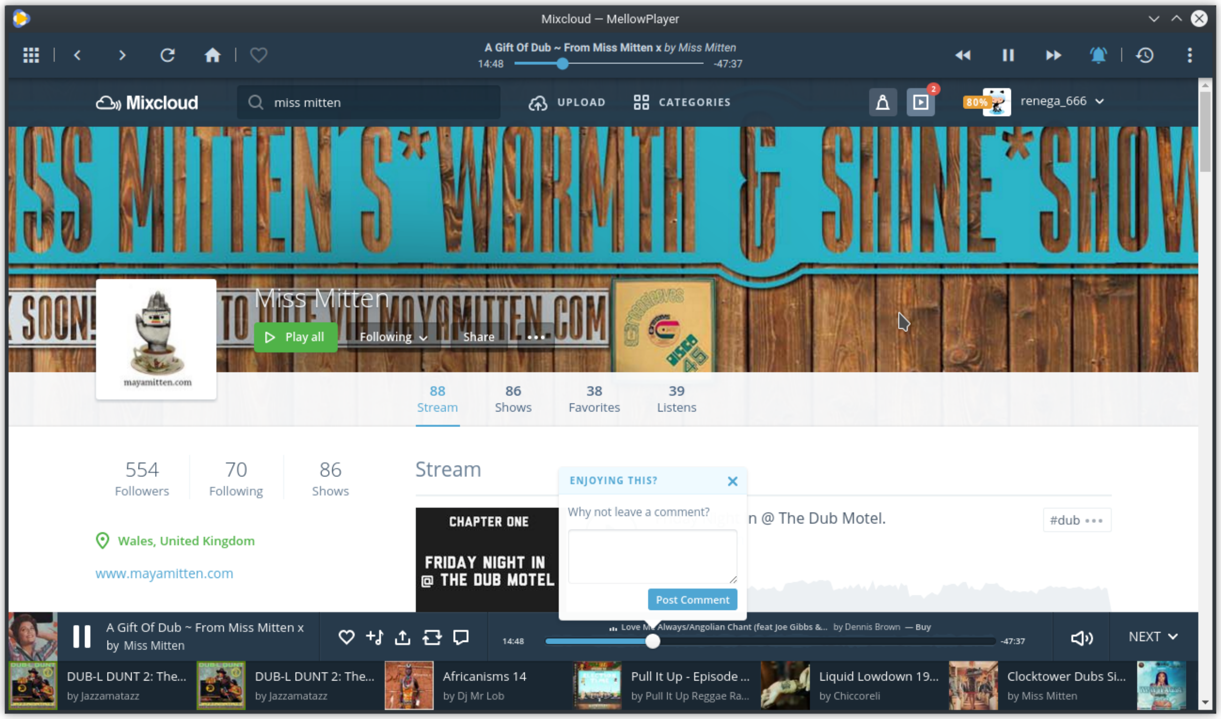

Mixcloud — MellowPlayer
Which streaming service would you like to use?
Show only favorite services
Search within available services
Deezer
Mixcloud
Soundcloud
TuneIn
Netflix
Youtube
Anghami
8tracks
amazon music

Just click on a service to start running it.

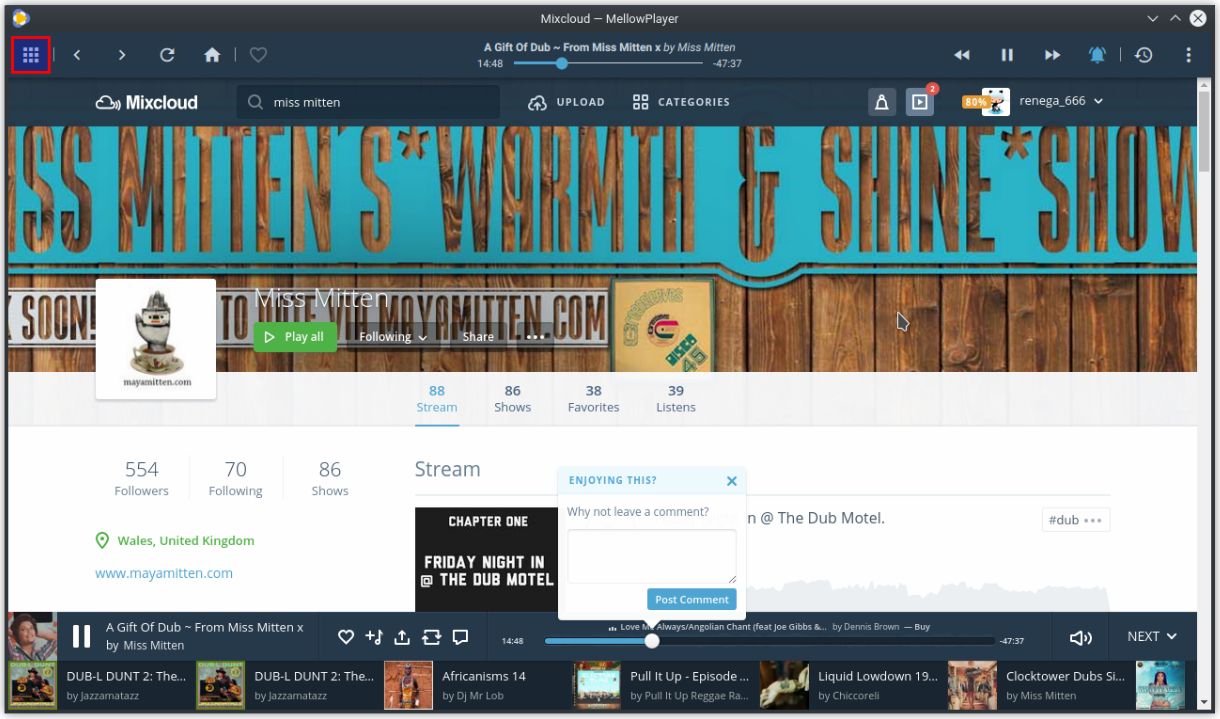
You can start multiple service at the same time and quickly switch between them using Ctrl+Tab/Ctrl+Shift+Tab. You can stop a service by clicking on the power button.

You can mark a service as a favorite by clicking on the little start icon. Favorite services are available from the system tray icon and you can choose to view only those services in the home page by activating the switch at the top left.

You can also use the filter bar to look for a service by name.

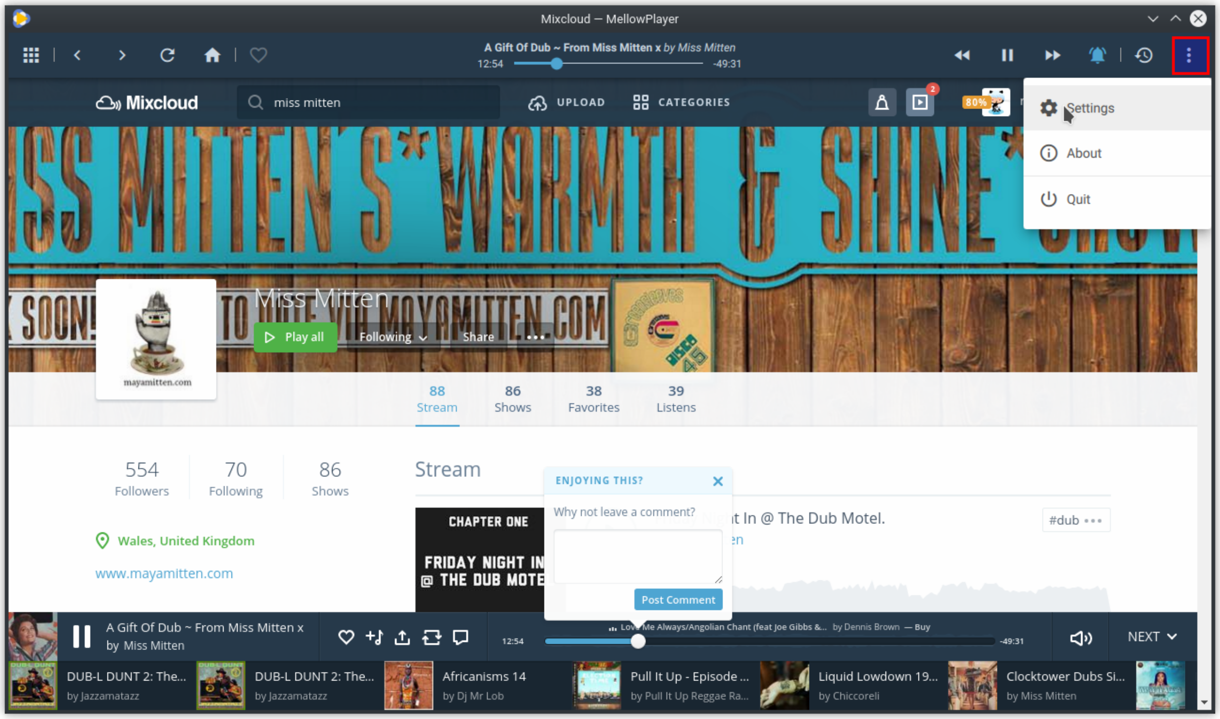


You can get back to the service selection page using F8 or the select service button:

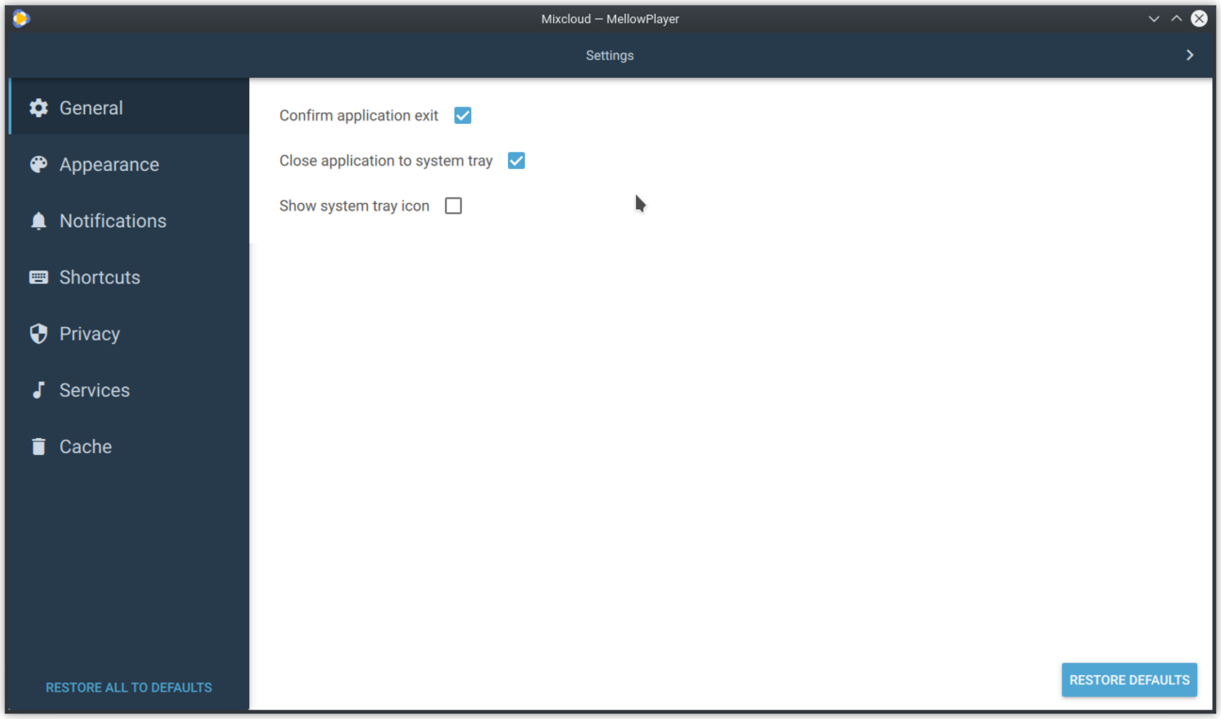


1.4.2 Application Settings

You can change application settings by clicking on the menu button and by selecting the Settings entry or by pressing F2:



This will bring the following page:



Confirm application exit ✓

Close application to system tray ✓

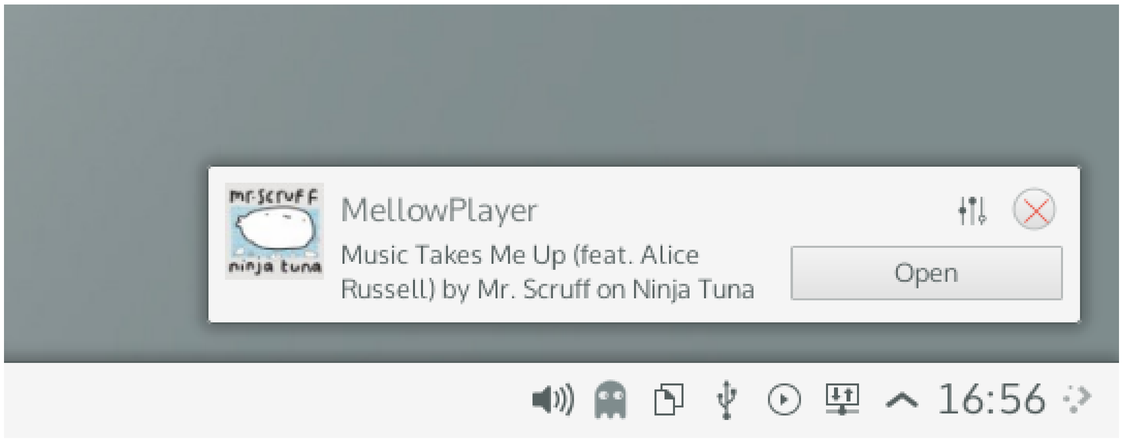
Show system tray icon ☐

There are a series of settings category:

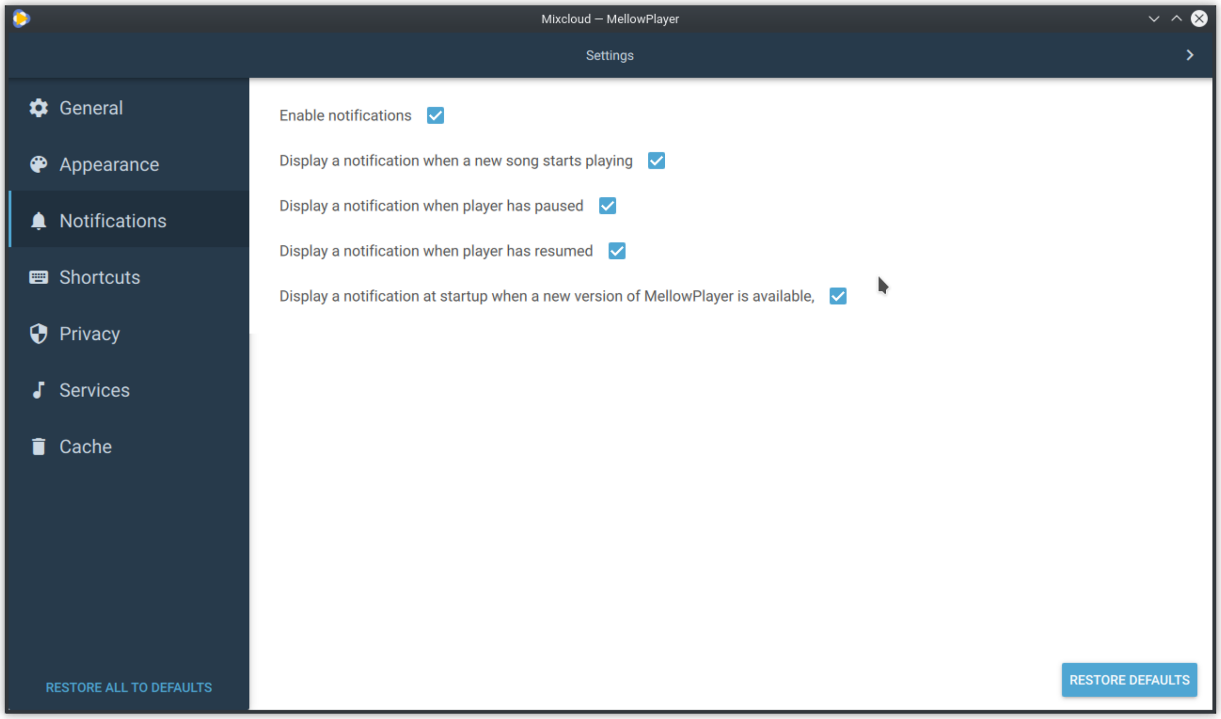
- General: general options
- Appearance: let you change the appearance of the application
- Notifications: let you change some notification settings
- Shortcuts: let you change all the application shortcuts
- Privacy: enable or disable privacy related options
- Services: list the available services and let configure them (url, user scripts)
- Cache: a few buttons that let you clear the application cache (album art covers,. . .) and clear the web cookies.

1.4.3 Notifications

By default, MellowPlayer will display a notification whenever the current track changed.


MellowPlayer
Music Takes Me Up (feat. Alice Russell) by Mr. Scruff on Ninja Tuna

You can change the notifications behaviour in the application settings page



Mixcloud – MellowPlayer

Settings

- General
- Appearance
- Notifications
- Shortcuts
- Privacy
- Services
- Cache

RESTORE ALL TO DEFAULTS

Enable notifications ✓

Display a notification when a new song starts playing ✓

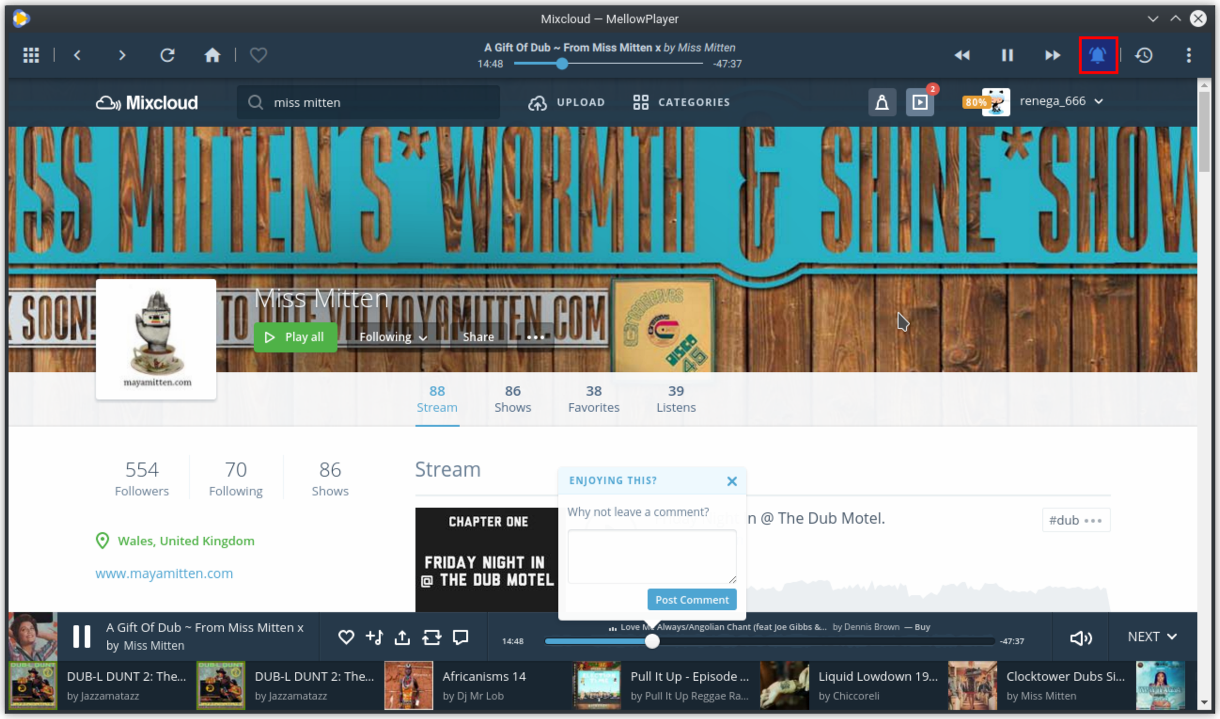
Display a notification when player has paused ✓

Display a notification when player has resumed ✓

Display a notification at startup when a new version of MellowPlayer is available, ✓

RESTORE DEFAULTS

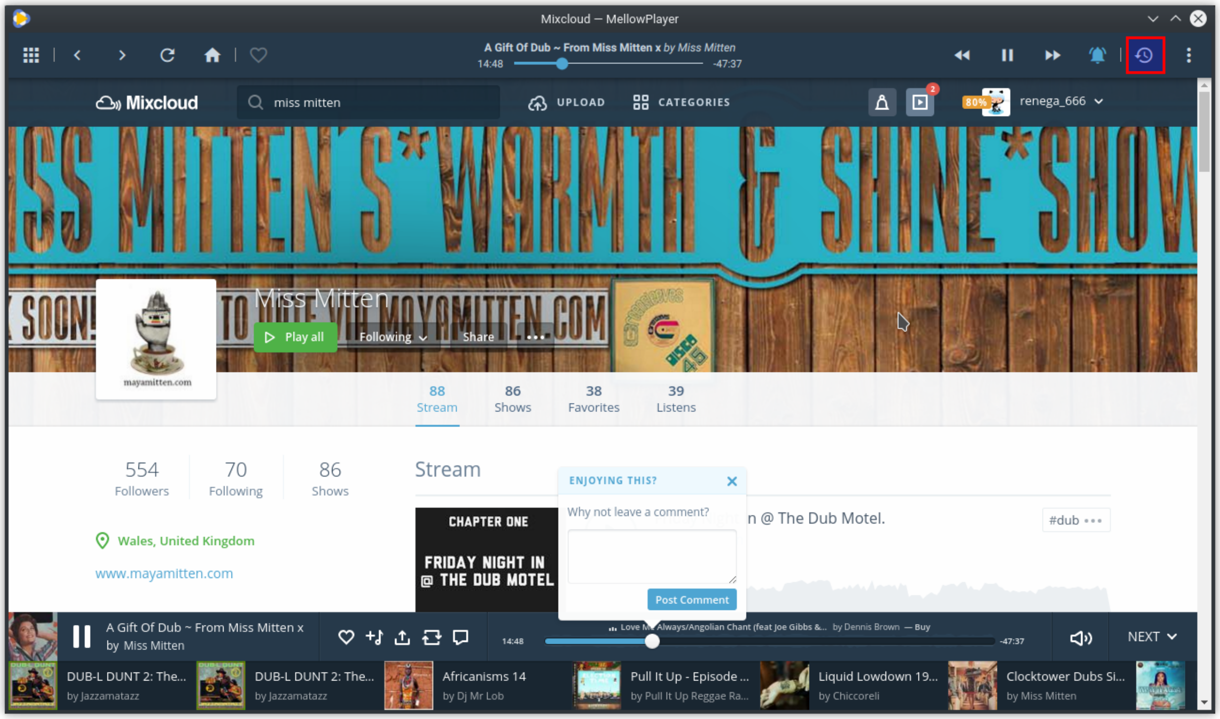
and you can also quickly toggle notifications on/off using the button in the toolbar:


Mixcloud – MellowPlayer
A Gift Of Dub ~ From Miss Mitten x by Miss Mitten
14:48
-47:37
Mixcloud
miss mitten
UPLOAD
CATEGORIES
renega_666
Miss Mitten
Play all
Following
Share
88 Stream
86 Shows
38 Favorites
39 Listens
554 Followers
70 Following
86 Shows
Wales, United Kingdom
www.mayamitten.com
Stream
ENJOYING THIS?
Why not leave a comment?
Post Comment
n @ The Dub Motel.
#dub
CHAPTER ONE
FRIDAY NIGHT IN @ THE DUB MOTEL
A Gift Of Dub ~ From Miss Mitten x by Miss Mitten
NEXT
DUB-L DUNT 2: The... by Jazzamatazz
DUB-L DUNT 2: The... by Jazzamatazz
Africanisms 14 by Dj Mr Lob
Pull It Up - Episode ... by Pull It Up Reggae Ra...
Liquid Lowdown 19... by Chiccoreli
Clocktower Dubs Si... by Miss Mitten

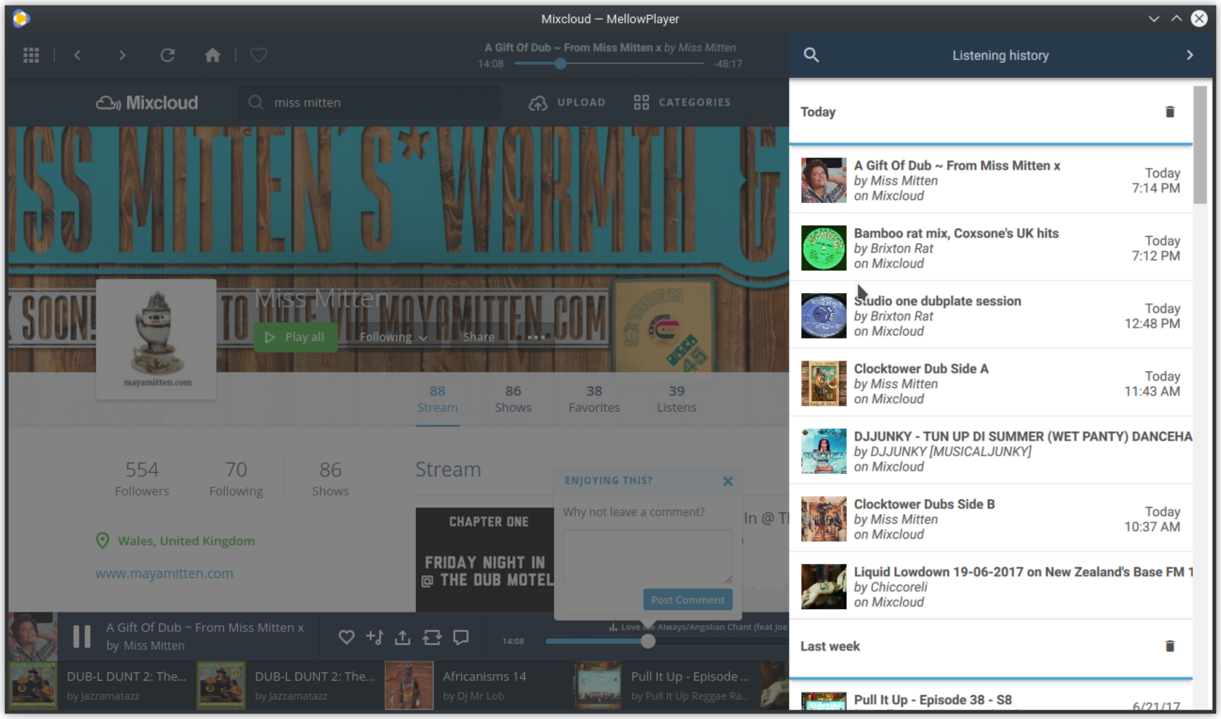
1.4.4 Listening History

MellowPlayer can keep track of your listens and display it in a side panel. This feature is OFF by default.

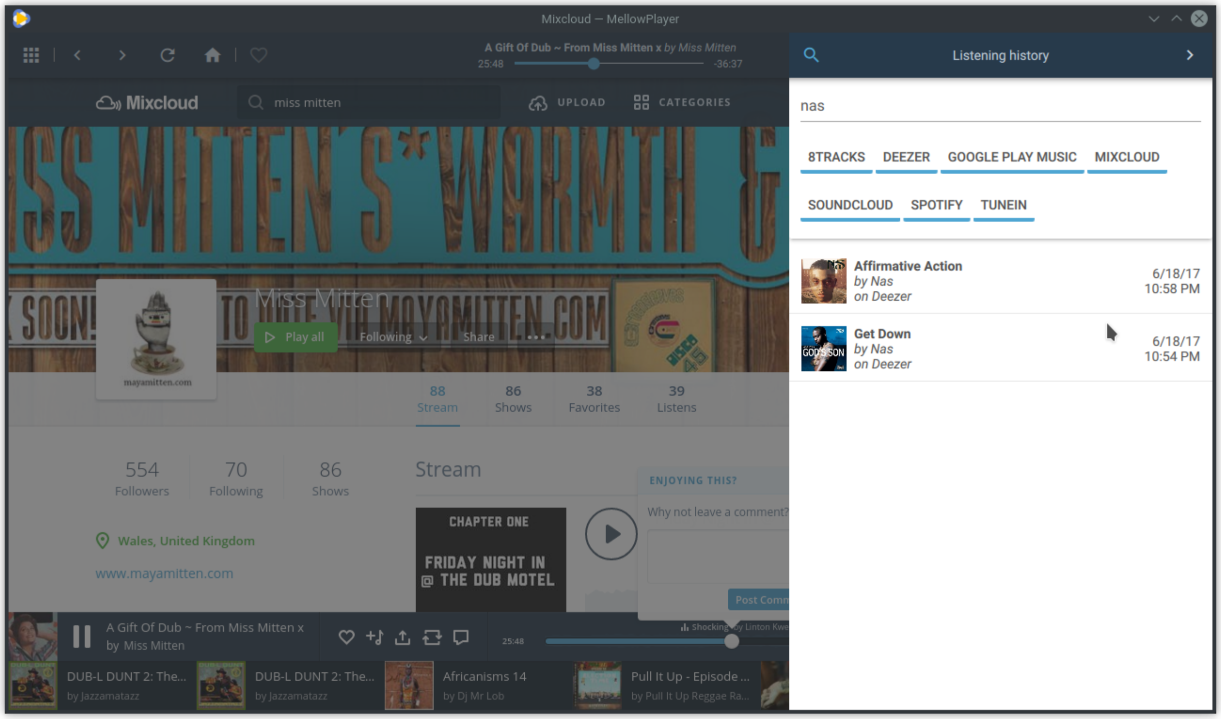
You can see your listening history by pressing the listening history button:



Here is what the history look like:


Mixcloud – MellowPlayer
Listening history
Today
A Gift Of Dub ~ From Miss Mitten x by Miss Mitten on Mixcloud Today 7:14 PM
Bamboo rat mix, Coxsone's UK hits by Brixton Rat on Mixcloud Today 7:12 PM
Studio one dubplate session by Brixton Rat on Mixcloud Today 12:48 PM
Clocktower Dub Side A by Miss Mitten on Mixcloud Today 11:43 AM
DJJUNKY - TUN UP DI SUMMER (WET PANTY) DANCEHA by DJJUNKY [MUSICALJUNKY] on Mixcloud
Clocktower Dubs Side B by Miss Mitten on Mixcloud Today 10:37 AM
Liquid Lowdown 19-06-2017 on New Zealand's Base FM 1 by Chiccoreli on Mixcloud
Last week
Pull It Up - Episode 38 - S8 6/21/17

You can search the history and filter by service by clicking on the search icon

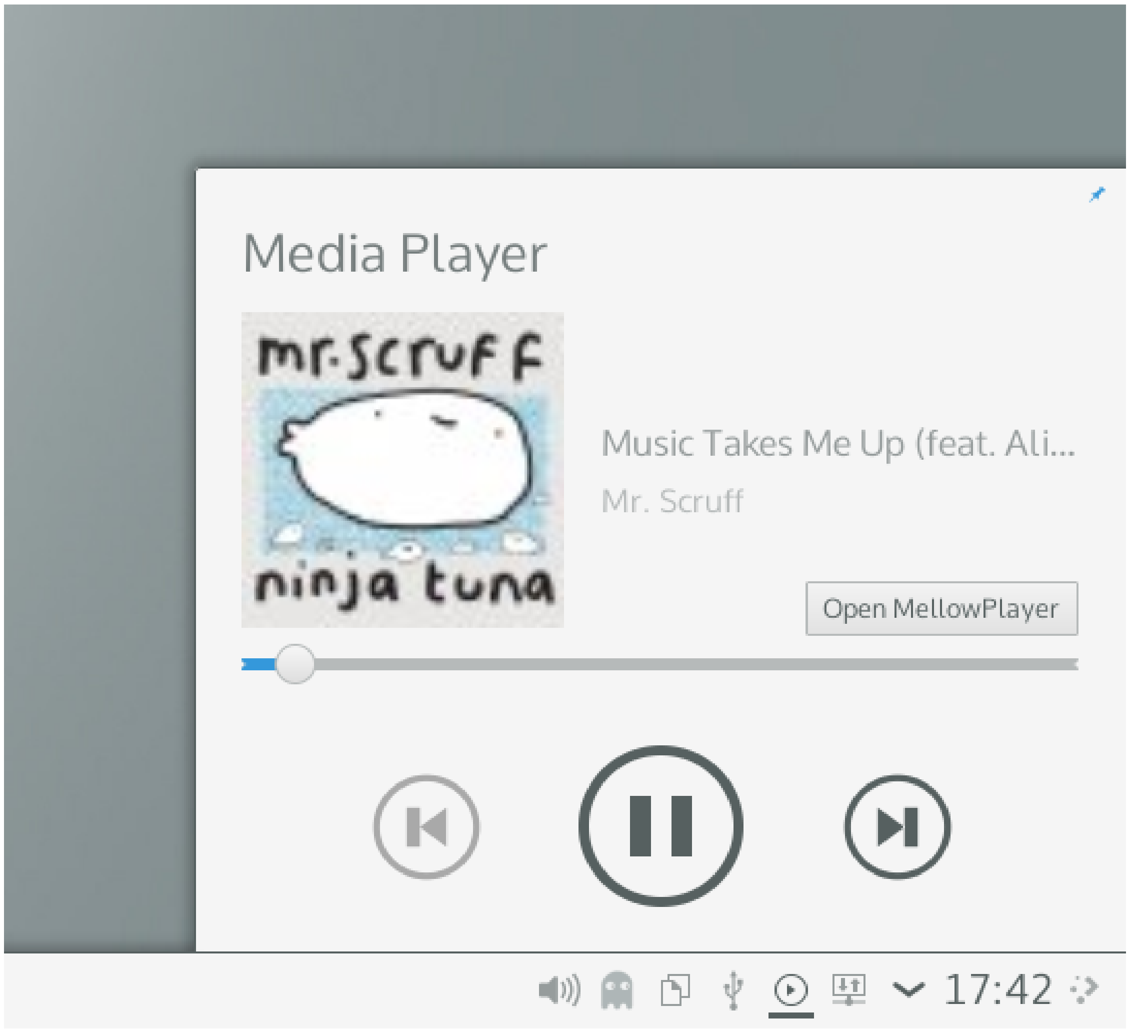


1.4.5 MPRIS2 Interface

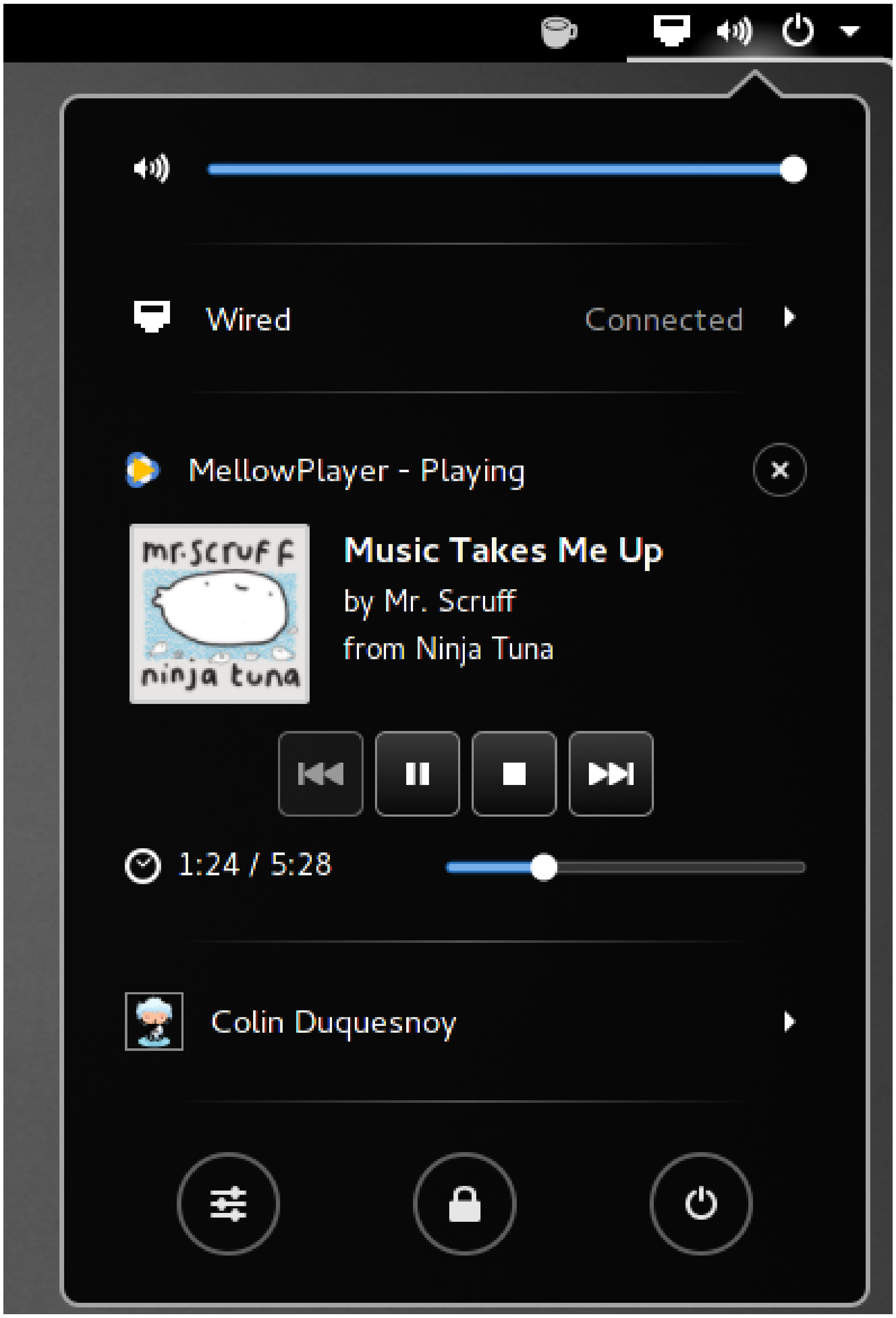
Most GNU/Linux Desktop Environments have a MPRIS client interface that sits in the system tray and let you control media players easily.

MellowPlayer implements the DBUS MPRIS 2 interface and should appear in your MPRIS2 client interface:

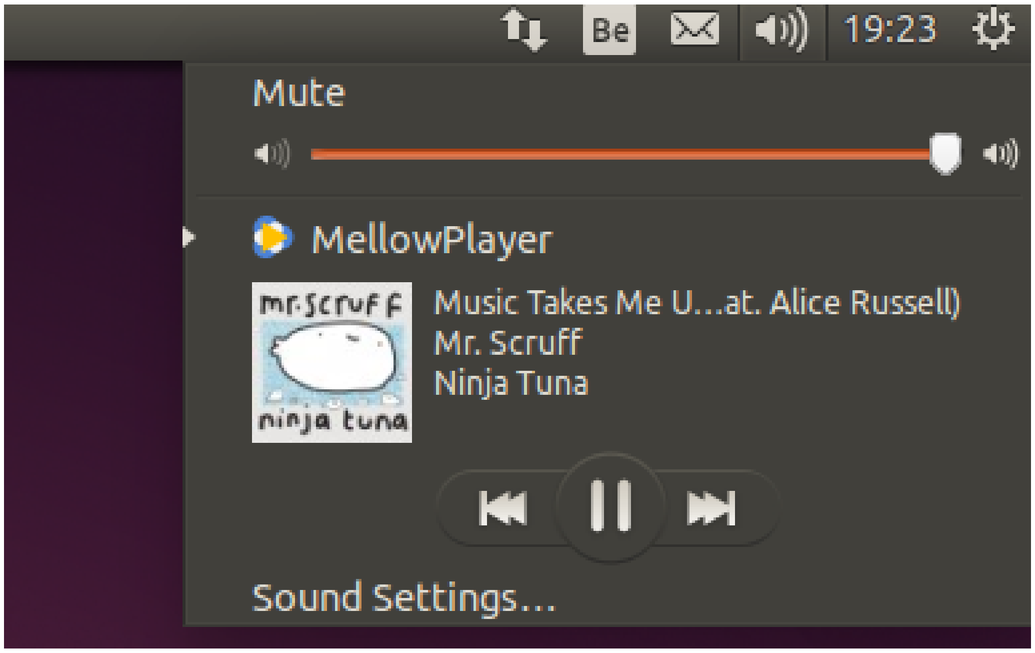
- Plasma 5:


Media Player
Music Takes Me Up (feat. Ali...
Mr. Scruff
Open MellowPlayer
17:42

- Gnome:


Wired
Connected
MellowPlayer - Playing
Music Takes Me Up
by Mr. Scruff
from Ninja Tuna
1:24 / 5:28
Colin Duquesnoy

- Unity:



1.4.6 Passing chromium flags

You can pass chromium flags using the **QTWEBENGINE_CHROMIUM_FLAGS** environemnt variable:

```
export QTWEBENGIN_CHROMIUM_FLAGS=--no-sandbox --disable-logging
./MellowPlayer
```

1.5 FAQ & Known issues

1.5.1 Playback does not start on some services… What can I do?

For many services to work, QtWebEngine/chromium needs to be **compiled** with support for proprietary Audio/Video codecs support (off by default in official pre-compiled Qt binaries) If you have playback issues on GNU/Linux, we recommend to use **'our flatpak'_** (which comes with all necessary codecs) instead of the native package or the AppImage.

Additionally, many services (Spotify, Tidal, Netflix, Amazon Music,…) use DRM and you **MUST** install the widevine DRM plugin.

1.5.2 Some services are not listed on Windows, is that normal?

Yes. The windows version of MellowPlayer is built with the official pre-compiled Qt binaries wich is built without proprietary codecs. Services that requires proprietary codecs are blacklisted on Windows becauouse they wouldn't work with our binary distribution anyway.

To workaround the isseu, try the following steps:

1. Edit the plugin metadata file ($INSTALL_DIR/plugins/PLUGIN_NAME/metadata.ini) and change the `support_platform` value to `All` instead of `Linux`
2. Recompile QtWebEngine from source and make sure to enable proprietary codecs support

3. Replace the QtWebEngine dll supplied by the MellowPlayer installer by the one you just built (or recompile MellowPlayer from source using your own version of Qt)
4. If the service require DRM, try to find the widevinecdm dll in your google chrom installation and copy it next to the MellowPlayer executable.

Note: We never tried the above mentioned steps as most of us are not Windows users. Your contribution is welcome!

1.5.3 The application crashes at startup on GNU/Linux with open source NVIDIA drivers. What can I do?

Qt/QML applications don't work well with the open source NVIDIA drivers (nouveau). It is recommended to **use the proprietary NVIDIA drivers**.

1.5.4 The application crashes at startup on GNU/Linux with proprietary NVIDIA drivers. What can I do?

Make sure you rebooted after your last NVIDIA driver update and make sure to run `sudo nvidia-xconfig` before reporting the issue.

1.5.5 There is a message saying that the browser is not supported or outdated. What can I do?

If you get the following (or similar) error message:

```
You are trying to sign in from a browser or app that doesn't allow us to keep your␣
↪account secure.
Try using a different browser.
```

Note: The message might a bit different (e.g. on Yandex, the message say the browser is outdated).

You may want to try to spoof your user agent with the one from Firefox.

To change the user agent in MellowPlayer: **Settings -> Privacy -> User Agent**.

1.5.6 My login credentials are lost or refused. What can I do?

If you're using a native version of MellowPlayer on an old distribution or our AppImage chances are the Qt (especially QtWebEngine/chromium) version is too old is not supported by the service you try to log in.

Use our flatpak instead.

1.5.7 I have a warning about broken integration plugin. What can I do?

Since version 3.6.0, MellowPlayer tries to detect broken plugins and display a message to warn the user.

Here are the circumstances under which such a warning may appear:

- there are some unhandled exception in the intergation plugin.

- there is a known open issue on our issue tracker with the “broken integration plugin” label.
- [not yet implemented] the song information is empty but the web page is playing audio

CHAPTER 2

Developer documentation

Contents:

2.1 Coding guidelines

We use the llvm code formating guidelines using clang-format.

We made a small script that will format any C++/javascript source file in the project to fit the style guidelines:

```
sh scripts/beautify.sh
```

To run this tool, you need to install the following packages:

- **clang-format**: `sudo pacman -S clang`
- **jsbeautifier**: `sudo pip3 install jsbeautifier`

Please, run this script before submitting a pull request!

See the coding guidelines wiki page for more information: https://gitlab.com/ColinDuquesnoy/MellowPlayer/wikis/coding-guidelines

2.2 Project structure

See the architecture wiki page for more information: https://gitlab.com/ColinDuquesnoy/MellowPlayer/wikis/architecture

2.3 Plugins

2.3.1 Introduction

MellowPlayer can be extended by writing a *streaming service integration plugin*.

A streaming service integration plugin is just a directoy that contains some specific files:

- **integration.js**: the actual code that integrates the service into MellowPlayer
- **logo.svg**: the logo of the service
- **metadata.ini**: plugin's metadata
- **theme.json**: optional theme definition. The colors defined in this file are used through the whole user interface if theme is set to **adaptive**.

The file **integration.js** contains a series of function that you must implement. Those functions will get called by the C++ application for updating the player state or when the user triggered an action (play, pause,. . .).

MellowPlayer will look for plugins in the following directories:

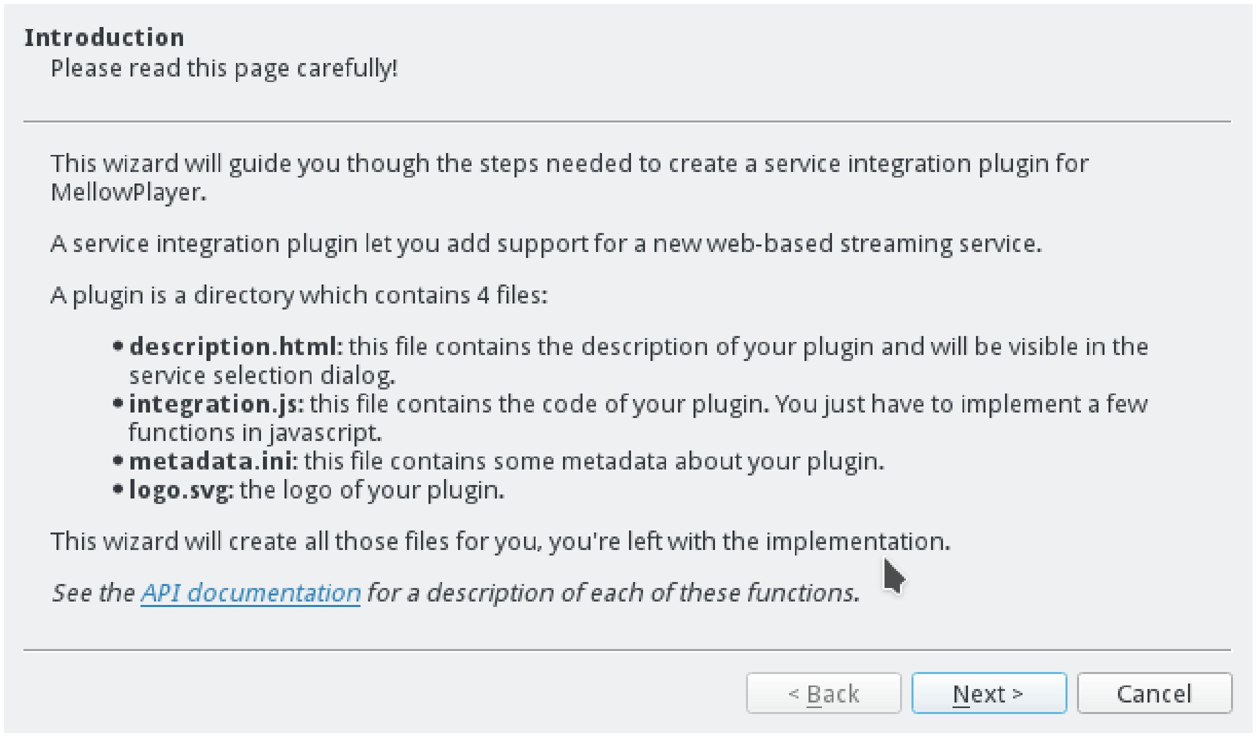
- **$CURRENT_WORKING_DIR/plugins**
- **/usr/share/mellowplayer/plugins**
- **/usr/local/share/mellowplayer/plugins**
- **~/.local/share/MellowPlayer/plugins**

2.3.2 Create a new plugin

This feature does not exists anymore in v2.95.0, we will be back for v3.0.0

To create a plugin, go to the **Control** drop down menu or the **Developer** main menu and click on **Create plugin**.

This will bring the following wizard:



Introduction
Please read this page carefully!

This wizard will guide you though the steps needed to create a service integration plugin for MellowPlayer.

A service integration plugin let you add support for a new web-based streaming service.

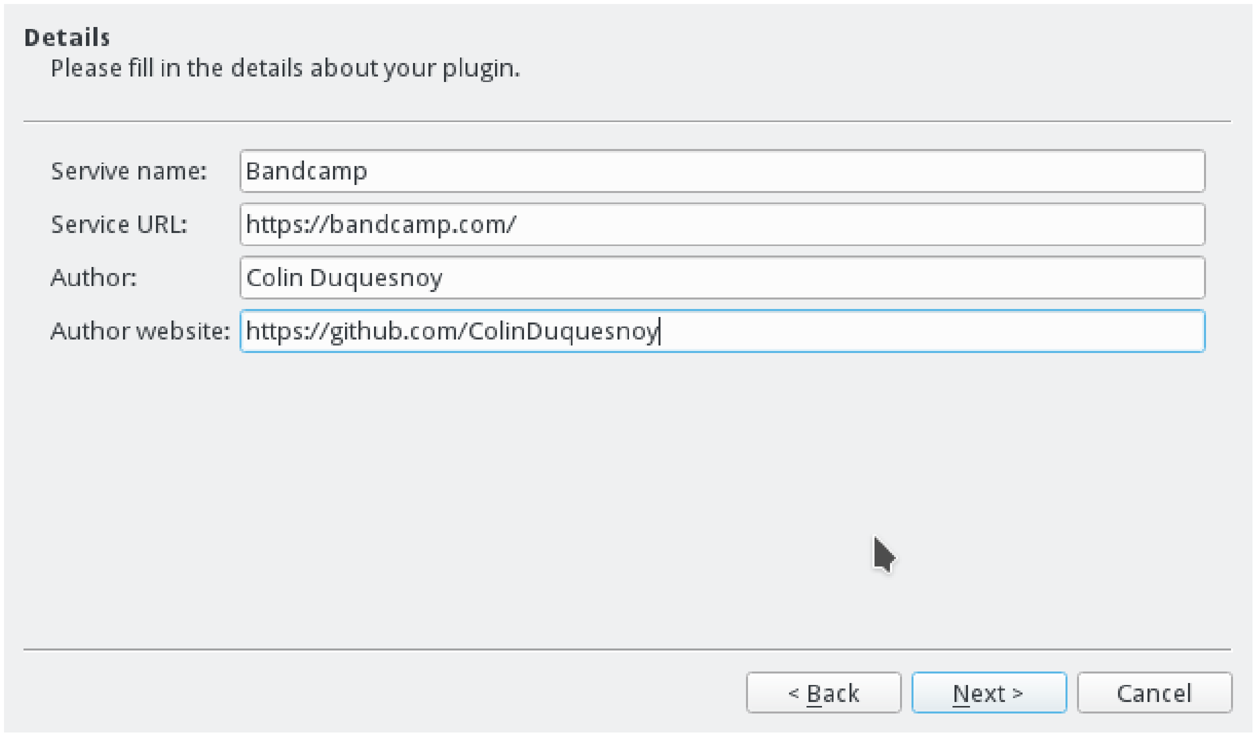
A plugin is a directory which contains 4 files:

- **description.html**: this file contains the description of your plugin and will be visible in the service selection dialog.
- **integration.js**: this file contains the code of your plugin. You just have to implement a few functions in javascript.
- **metadata.ini**: this file contains some metadata about your plugin.
- **logo.svg**: the logo of your plugin.

This wizard will create all those files for you, you're left with the implementation.

See the API documentation for a description of each of these functions.

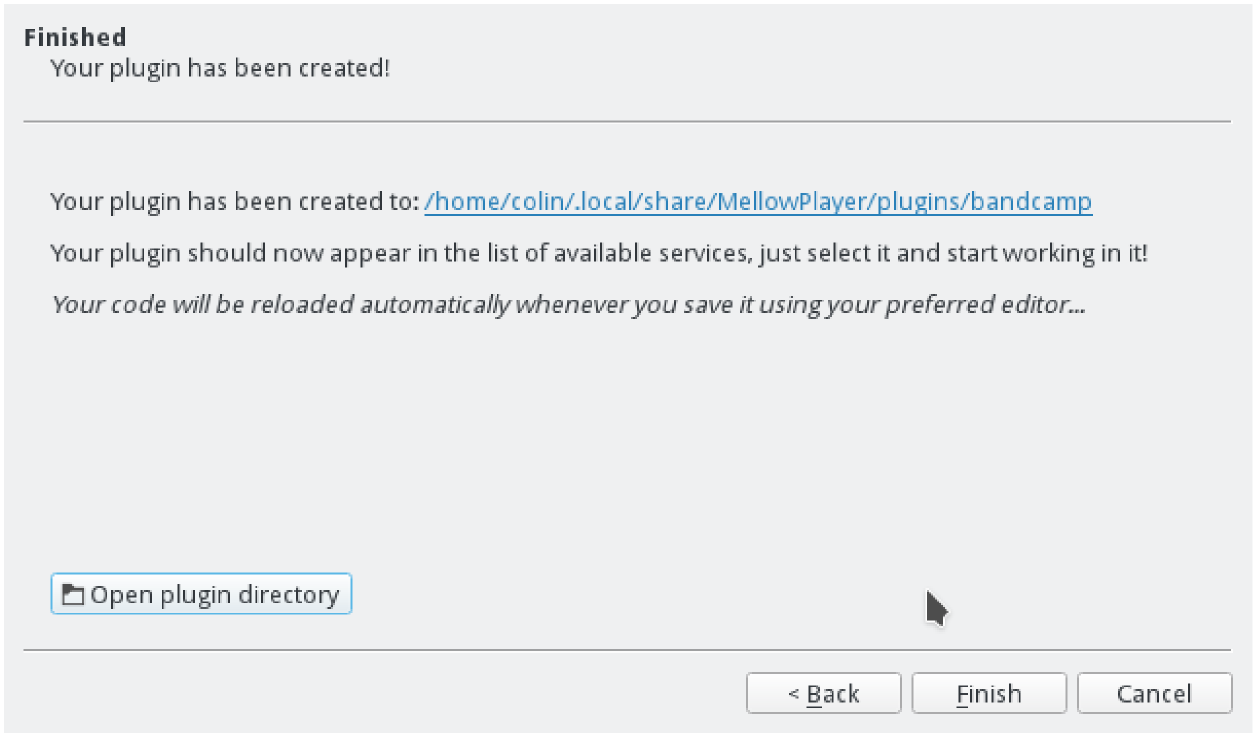
Fill in the details:



Details
Please fill in the details about your plugin.

Servive name:	Bandcamp
Service URL:	https://bandcamp.com/
Author:	Colin Duquesnoy
Author website:	https://github.com/ColinDuquesnoy

When you're done, select your new plugin service in the services dialog that will automatically pop out:


Finished

Your plugin has been created!

Your plugin has been created to: /home/colin/.local/share/MellowPlayer/plugins/bandcamp

Your plugin should now appear in the list of available services, just select it and start working in it!

Your code will be reloaded automatically whenever you save it using your preferred editor...

2.3.3 Specify the supported platforms

When you create the plugin, you need to specify the list of supported platforms. Services that require proprietary codecs to work are not supported on Windows and OSX.

2.3.4 Functions to implement

Here is a brief description of the functions you need to implement in order to integrate a new web-based streaming service.

update()

This function is called regularly to update the player information.

You must return a dictionnary with the following keys:

- **playbackStatus** *(int)*: Use MellowPlayer.PlaybackStatus)*. **Mandatory**
- **canSeek** *(bool)*: True if the player can seek into the current song.
- **canGoNext** *(bool)*: True if the player can skip to the next song.
- **canGoPrevious** *(bool)*: True if the playe can skip to the previous song.
- **canAddToFavorites** *(bool)*: True if the user can add or remove the current song from a list of favorites
- **volume** *(float [0-1])*: Player volume. Optional, leave it 1 if your plugin cannot control the volume.
- **songId** *(str)*: The unique id of the current song. **Mandatory**. Either use a GUID or hash the song title if no id is available.
- **songTitle** *(str)*: The title of the current song. **Mandatory**

- **artistName** *(str)*: The name of the artist of the current. Optional.
- **albumTitle** *(str)*: The name of the album of the current song. Optional.
- **artUrl** *(str)*: The current song art url.
- **isFavorite** *(bool)*: True if the song is in the list of the user's favorite songs. Optional.
- **duration** *(int [seconds])*: The duration of the song, in seconds. Optional (but nice).
- **position** *(int [seconds])*: The position (or elapsed time) of the song, in seconds. Optional (but nice).

play()

Starts playback.

pause()

Pauses playback.

goNext()

Skips to next song.

goPrevious()

Skips to previous song.

setVolume(volume)

Sets the player's volume.

volume is a *float* in the range [0-1].

addToFavorites()

Adds song to favorites.

removeFromFavorites()

Removes song from favorites.

seekToPosition(position)

Seeks to the specified position.

position is an *int* representing the new position inside the song (in seconds).

2.3.5 PlaybackStatus

MellowPlayer will inject a few constants that you can use for representing the current PlaybackStatus:

- **MellowPlayer.PlaybackStatus.STOPPED**: indicates that the playback has stopped.
- **MellowPlayer.PlaybackStatus.PAUSED**: indicates that the playback has paused.
- **MellowPlayer.PlaybackStatus.BUFFERING**: indicates that the a song is buffering.
- **MellowPlayer.PlaybackStatus.PLAYING**: indicates that the a song is currently playing.

2.3.6 Utility functions

- `function getHashCode(string)`: returns the hash code of the specified string. You can use this to generate the song id if none is available via the web streaming service API.
- `toSeconds(string)` converts a time string (`HH:mm:ss`) to a number of seconds.

2.4 Contributing to MellowPlayer

2.4.1 Reporting bugs or Wishes

Report any bugs you encountered or any wishes on our issue tracker.

If you're reporting a bug, **make sure to provide the following information**:

- Information about your **Operating system** (e.g. Windows 8.1, Mac OSX Yosemite,. . .). If you're on Linux, you'll need to specify the name of the distribution and the desktop environment you're using and whether you're using a native package or the flatpak.
- The **music streaming service** that you were using when you encountered the bug if related to a specific streaming service.
- A **clear description** of the bug with **steps to reproduce**.
- You should use **English** to describe your issue. French is also accepted.
- Paste the application log between triple backquotes `(About > Show Logs).`

2.4.2 Setting up a development environment

Read the how to setup page of the wiki

Some helper scripts to setup your development environment for different GNU/Linux distributions can be found in the scripts/env-setup folder.

We also recommend you read the architecture and the coding guidelines pages before hacking on MellowPlayer.

2.4.3 Submitting a pull request

Here are the steps you need to follow to start working on MellowPlayer and submit your work for evaluation or integration into the main project:

1. Fork the Repo on gitlab.
2. Create a feature or a bugfix branch before you start coding.

3. Add your name to AUTHORS.md
4. Format the code using `scripts/beautify.py` (run it from the root source directory).
5. Push to your fork and submit a pull request to **the develop branch**.

2.4.4 Adding support for a new service

Web streaming service integration plugins are now written in pure javascript.

1. Create a new plugin using the wizard (see http://mellowplayer.readthedocs.io/en/latest/developers/plugins.html#create-a-new-plugin)
2. Edit `metadata.ini` (add correct url, name, version,. . .)
3. Edit `description.html` to describe the streaming service
4. Customise logo.svg
5. Implement the needed functions in `integration.js`
6. Once your plugin works, submit a pull request to **the develop branch**.

2.4.5 Adding/Updating a new translation

MellowPlayer translations are hosted on transifex

- Create an account at transifex
- Go to the project's homepage and click on the "Join the team" button
- If the language you want to work on does not exists yet, send us a language request. Once the request has been accepted, a new translation file for the requested language will be created automatically by transifex.
- To actually start translating, go to the project's home page on transifex and click on the tr

CHAPTER 3

Indices and tables

- genindex
- modindex
- search