

# Native Spotify

Describes how to listen to songs from Spotify on SailfishOS with Hutspot and Librespot.

DISCLAIMER: This only works with Spotify Premium accounts. A free Spotify subscription will not work with these applications

- [Hutspot](#)
- [Use Librespot to play songs](#)
- [Help links](#)

# Hutspot

## Overview

Hutspot is a native SailfishOS-app that acts as a Spotify-controller. It enables you to browse/search albums, artists, playlists and tracks. Playing the music is done on a 'connected' device. It requires a premium Spotify account.

## Documentation

Documentation for the app can be found on <https://sailfish-spotify.github.io/hutspot/> . The source code is also accessible under <https://github.com/sailfish-spotify/hutspot>

## Installation

Hutspot is available in the Chum-repositories. To see it, you need to disable "Show only applications" in the Chum GUI app.

## Fix authentication issue

The current version in Chum has an issue with the authentication, where you would need to authenticate every time you launch the app. To fix this issue, use a terminal app to copy the *hutspot.desktop* file to */etc/sailjail/applications*:

```
cp /usr/share/applications/hutspot.desktop /etc/sailjail/applications
```

After that, use a text editor (for example `nano`) to add the following lines to the file:

```
[X-Sailjail]
Sandboxing=Disabled
```

# Use Librespot to play songs

## Overview

*librespot* is an open source client library for Spotify. It enables applications to use Spotify's service to control and play music and to act as a Spotify Connect receiver. Using this terminal-application in combination with a Spotify-controller like [Hutspot](#), you can play songs from Spotify on your SailfishOS-Device without using the Android-app from Spotify itself. The source code is available on [GitHub](#)

## Installation

### Prebuilt package

**ATTENTION:** Currently (11.7.2024) this package DOES NOT work on SailfishOS 4.6 due to some dependency problem. If you want to use Librespot on this SailfishOS Version, you need to build it yourself (see tutorial below)

There is a prebuilt package which a user uploaded to the [forum](#). You can download it [here](#). If you want to change the configuration, you can edit the command in the file `/usr/lib/systemd/user/librespot.service`. This package installs Librespot as a service, so you can start and stop it from Hutspot directly.

### Building it yourself

You can also get librespot directly and build it on your device. First, you need to install Rust. To do this, open a terminal and execute

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

After the installation has finished, you need to restart your session. This can be done by just typing `bash` and hitting *enter*. You also need to install a few development packages. To install these, type in a terminal

```
devel-su pkcon refresh  
devel-su pkcon install gcc git alsa-lib-devel -y
```

Now that you have everything necessary installed, you can get the source code from GitHub

```
git clone https://github.com/librespot-org/librespot  
cd librespot/
```

After that, you can now build your application using `cargo`:

```
cargo build --release --features "alsa-backend pulseaudio-backend"
```

# Usage

## Controlling Librespot with Hutspot

If you have installed Librespot as a service, you can control Librespot right from the GUI of Hutspot. To do this, tap on the *Hamburger-menu* (four lines) in the bottom right corner of Hutspot and navigate to *Settings*. There, you need to activate "Control Librespot" (Like seen in [this screenshot](#)). This will start Librespot with Hutspot and terminate it when Hutspot is closed. On this page, you can also start/stop Librespot manually. This is also possible from the Pull-Down-Menu of the *Devices*-page (Like seen [here](#)).

## Starting Librespot from the terminal

You can also start Librespot from the terminal, where you can make your configuration directly. If you have built Librespot on your own, this is the only option to start Librespot. In this case, you need to execute Librespot from the path where you built it. The executable is typically stored under `<PATH TO LIBRESPOT SOURCE>/target/release/`. For more overview of which options are possible for Librespot, you can look [here](#).

Generally, you would want the `--username` option, the `--cache` option (so you don't need to type in your password everytime), the `-n` option to give the instance a name and the `--backend` option (which needs to be set to `pulseaudio`):

An example command could look like this:

```
/home/defaultuser/librespot/target/release/librespot --cache /home/defaultuser/.cache/librespot --username  
USERNAME -b 320 -n MyAwesomeLibrespotBuild --backend pulseaudio --device-type smartphone
```

## Setting Librespot as the current device

If Librespot is running, you can use it to play songs from Spotify via Hutspot. To do this, open Hutspot and navigate to the *Devices*-page. All connected Spotify devices will show up there (which includes Librespot-instances). To use a device, long press on the device and choose "Set as active

device" (first option in the menu). Congratulations, you can now play, pause and skip songs via Hutspot which will be played using your SailfishOS-smartphone.

**Notice:** The volume of Librespot can't be controlled with the volume-buttons on your device. For that, you need to use the volume-slider in Hutspot.

[Bildschirmfoto\\_20230923\\_002.png](#)

# Help links

If you need help, here are a few useful links to various posts in the SailfishOS-Forum:

- <https://forum.sailfishos.org/t/hutspot-for-aarch64/8231>
- <https://forum.sailfishos.org/t/cargo-install-error-this-is-the-stub-cargo-if-you-see-this-scratchbox2-is-not-working/11478>