
librespot-python

kokarare1212

Aug 05, 2023

CONTENTS

| | | |
|----------|---------------------------------|-----------|
| 1 | Contents | 3 |
| 2 | What’s librespot-python? | 5 |
| 3 | What do you need? | 7 |
| 4 | Disclaimer | 9 |
| 5 | Installation | 11 |
| 6 | Usage | 13 |
| 7 | Debug | 15 |

CONTENTS

1.1 Supported Futures

| Futures | Supported |

| — | — |

| Get Metadata | |

| Play Music | *1 |

| Spotify Connect | ×*2 |

*1 It is currently possible to retrieve the music stream, but it requires separate software for decoding.

*2 This feature will be added in the future.

1.2 API Reference

WHAT'S LIBRESPOT-PYTHON?

librespot-python is a python port of Spotify's open source client library [librespot](#). It was created to play Spotify on various platforms and devices.

WHAT DO YOU NEED?

In order to develop with this library, you need to use Python. Python can be downloaded from [the official website here](#).

DISCLAIMER

Please keep in mind that this library is not like Spotify approved. Therefore, the worst that can happen is that you will be banned from Spotify. Also, please keep in mind that this library is in alpha and may behave in unintended ways.

INSTALLATION

You can download this library in the following way. Stable Version

```
pip install librespot
```

Snapshot Version ***Recommended**

```
pip install git+https://github.com/kokarare1212/librespot-python
```


6.1 Get Spotify's OAuth token

```
from librespot.core import Session

session = Session.Builder() \
    .user_pass("Username", "Password") \
    .create()

access_token = session.tokens().get("playlist-read")
```

6.2 Get Music Stream

*Currently, music streaming is supported, but it may cause unintended behavior.

```
from librespot.core import Session
from librespot.metadata import TrackId
from librespot.audio.decoders import AudioQuality, VorbisOnlyAudioQuality

session = Session.Builder() \
    .user_pass("Username", "Password") \
    .create()

track_id = TrackId.from_uri("spotify:track:xxxxxxxxxxxxxxxxxxxxxx")
stream = session.content_feeder().load(track_id, VorbisOnlyAudioQuality(AudioQuality.
    ↪VERY_HIGH), False,
                                None)

# stream.input_stream.stream().read() to get one byte of the music stream.
# ex: 1 (If there is no more voice data, -1 is received as the result.)
```


DEBUG

To display the debug information, you need to inject the following code at the top of the code.

```
import logging

logging.basicConfig(level=logging.DEBUG)
```