
Grooveshark API Documentation

Release 0.1

Rafael Soares dos Santos

Jun 13, 2017

Contents

1	Quickstart	3
1.1	Creating a Grooveshark instance to have a user's data with Username and Password.	3
1.2	Creating a Grooveshark instance to have a user's data with Session ID.	4
1.3	After you have the Grooveshark's instance, just get the data. (Example)	4
1.4	If you do not want to have a user's data. (Example)	4
1.5	Supported methods by now	4

New in version 0.2: Changes in the structure's module. Before, you should import Grooveshark from 'grooveshark_api.grooveshark', now you need to import from 'grooveshark_api'. The method 'get_stream_key_stream_server' now accept the same arguments as the official Grooveshark API, as described in the website. Added more methods.

CHAPTER 1

Quickstart

This is a Wrapper of the Grooveshark Public API (v3). See [Grooveshark API V3](#).

Just create a Grooveshark instance, and access the API Methods in a pythonic way.

Note: 'getDoesSongExist' will be 'get_does_song_exist'

Creating a Grooveshark instance to have a user's data with Username and Password.

```
• import hashlib
  from grooveshark_api import Grooveshark

GS_KEY = 'GS_KEY'
GS_SECRET = 'GS_SECRET'
user_login = 'username'
user_pass = 'password'

md5 = hashlib.md5()
md5.update(user_pass)
user_pass = md5.hexdigest()

gs = Grooveshark(key=GS_KEY,
                  secret=GS_SECRET,
                  user_login=user_login,
                  user_pass=user_pass)
```

Note: Session ID is used to login a user again. Just keep his Session ID, and use it to make his authentication in the next time. See the official Grooveshark API docs to get more information.

To have the Session ID:

```
print gs.sessionID
```

Creating a Grooveshark instance to have a user's data with Session ID.

```
from grooveshark_api import Grooveshark

GS_KEY = 'GS_KEY'
GS_SECRET = 'GS_SECRET'
SESSIONID = 'SESSIONID'

gs = Grooveshark(key=GS_KEY,
                 secret=GS_SECRET,
                 sessionID=SESSIONID)
```

After you have the Grooveshark's instance, just get the data. (Example)

```
# some code here

user_playlists = gs.get_user_playlists(limit=10)
```

If you do not want to have a user's data. (Example)

```
# some code here

gs = Grooveshark(key=GS_KEY, secret=GS_SECRET)
popular_songs = gs.get_popular_songs_today()
```

Supported methods by now

- `get_user_playlists(self, limit=None)`
- `get_playlist(self, playlist_id, limit=None)`
- `get_does_song_exist(self, song_id)`
- `get_songs_info(self, songs_id)`
- `get_stream_key_stream_server(self, song_id, country, low_bitrate=False)`
- `get_country(self, ip=None)`
- `get_playlist_search_results(self, query, limit=None)`

- `get_album_search_results(self, query, limit=None)`
- `get_song_search_results(self, query, country, limit=None, offset=None)`
- `get_artist_search_results(self, query, limit=None)`
- `get_autocomplete_search_results(self, query, query_type, limit=None)`
- `add_user_library_songs(self, song_ids, album_ids, artist_ids)`
- `get_user_library_songs(self, limit=None, page=None)`
- `add_user_library_songs_ex(self, songs)`
- `remove_user_library_songs(self, song_ids, album_ids, artist_ids)`
- `get_user_playlists_subscribed(self)`
- `get_user_favorite_songs(self, limit=None)`
- `remove_user_favorite_songs(self, song_ids)`
- `logout(self)`
- `get_user_info(self)`
- `get_user_subscription_details(self)`
- `add_user_favorite_song(self, song_id)`
- `subscribe_playlist(self, playlist_id)`
- `unsubscribe_playlist(self, playlist_id)`
- `get_playlist_info(self, playlist_id)`
- `get_popular_songs_today(self, limit=None)`
- `get_popular_songs_month(self, limit=None)`
- `ping_service(self)`
- `get_service_description(self)`
- `undelete_playlist(self, playlist_id)`
- `delete_playlist(self, playlist_id)`
- `get_playlist_songs(self, playlist_id, limit=None)`
- `set_playlist_songs(self, playlist_id, song_ids)`
- `create_playlist(self, name, song_ids)`
- `rename_playlist(self, playlist_id, name)`
- `get_user_id_from_username(self, username)`
- `get_albums_info(self, album_ids)`
- `get_album_songs(self, album_id, limit=None)`
- `get_artists_info(self, album_ids)`
- `get_does_album_exist(self, album_id)`
- `get_does_artist_exist(self, artist_id)`
- `get_artist_verified_albums(self, artist_id)`
- `get_artist_albums(self, artist_id)`

- `get_artist_popular_songs(self, artist_id)`