

---

**google-music**

*Release 3.7.0*

**thebigmunch**

**Oct 14, 2020**



# CONTENTS

<b>1</b>	<b>Getting Started</b>	<b>3</b>
1.1	API — High-Level API . . . . .	3
1.2	MobileClient — Mobile API client . . . . .	4
1.3	MusicManager — Music Manager API client . . . . .	21
1.4	Sessions — Google Music HTTP Sessions . . . . .	22
1.5	Token Handlers — OAuth Token Handlers . . . . .	23
	<b>Index</b>	<b>25</b>



**Due to Google Music shutting down in favor of YouTube Music, this project has ended.\*\*\*Due to Google Music shutting down in favor of YouTube Music, this project has ended.**



## GETTING STARTED

Install google-music with `pip`.

```
$ pip install google-music
```

Create a mobile or music manager client using the high-level API.

```
>>> import google_music
>>> mc = google_music.mobileclient()
>>> mm = google_music.musicmanager()
```

### 1.1 API — High-Level API

**mobileclient** (*username=None, device\_id=None, \*, locale='en\_US', session=None, token=None, token\_handler=<class 'google\_music.token\_handlers.FileTokenHandler'>, token\_handler\_kwargs=None*)

Create and authenticate a Google Music mobile client.

```
>>> import google_music
>>> mc = google_music.mobileclient('username')
```

#### Parameters

- **username** (*str, Optional*) – Your Google Music username. Used to store OAuth tokens for multiple accounts separately.
- **device\_id** (*str, Optional*) – A mobile device ID. Default: MAC address is used.
- **locale** (*str, Optional*) – ICU locale used to localize some responses. This must be a locale supported by Android. Default: 'en\_US'.
- **session** (*GoogleMusicSession, Optional*) – A session compatible with `GoogleMusicSession`.
- **token** (*dict, Optional*) – An OAuth token compatible with `oauthlib`.
- **token\_handler** (*TokenHandler, Optional*) – A token handler class compatible with `TokenHandler` for dumping and loading the OAuth token.
- **token\_handler\_kwargs** (*dict, Optional*) – Keyword arguments to pass to the `token_handler` class. These become attributes on the class instance.

**Returns** An authenticated `MobileClient` instance.

Return type *MobileClient*

```
musicmanager (username=None, uploader_id=None, *, session=None, token=None, to-  
ken_handler=<class 'google_music.token_handlers.FileTokenHandler'>, to-  
ken_handler_kwargs=None)
```

Create and authenticate a Google Music Music Manager client.

```
>>> import google_music  
>>> mm = google_music.musicmanager('username')
```

#### Parameters

- **username** (*str*, *Optional*) – Your Google Music username. Used to store OAuth tokens for multiple accounts separately.
- **uploader\_id** (*str*, *Optional*) – A unique uploader ID. Default: MAC address and username used.
- **session** (*GoogleMusicSession*, *Optional*) – A session compatible with *GoogleMusicSession*.
- **token** (*dict*, *Optional*) – An OAuth token compatible with *oauthlib*.
- **token\_handler** (*TokenHandler*, *Optional*) – A token handler class compatible with *TokenHandler* for dumping and loading the OAuth token.
- **token\_handler\_kwargs** (*dict*, *Optional*) – Keyword arguments to pass to the *token\_handler* class. These become attributes on the class instance.

**Returns** An authenticated *MusicManager* instance.

Return type *MusicManager*

## 1.2 MobileClient — Mobile API client

```
class MobileClient (username=None, device_id=None, *, locale='en_US',  
session=None, token=None, token_handler=<class  
'google_music.token_handlers.FileTokenHandler'>, to-  
ken_handler_kwargs=None)
```

API wrapper class to access Google Music mobile client functionality.

```
>>> from google_music import MobileClient  
>>> mc = MobileClient('username')
```

---

**Note:** Streaming requires a *device\_id* from a valid, linked mobile device. The *MobileClient* instance's *device\_id* can be changed after instantiation, or a different *device\_id* provided to *stream* or *stream\_url*.

---

#### Parameters

- **username** (*str*, *Optional*) – Your Google Music username. Used to store OAuth tokens for multiple accounts separately.
- **device\_id** (*str*, *Optional*) – A mobile device ID. Default: MAC address is used.

- **locale** (*str*, *Optional*) – ICU locale used to localize some responses. This must be a locale supported by Android. Default: 'en\_US'.
- **session** (GoogleMusicSession, *Optional*) – A session compatible with GoogleMusicSession.
- **token** (*dict*, *Optional*) – An OAuth token compatible with oauthlib.
- **token\_handler** (TokenHandler, *Optional*) – A token handler class compatible with TokenHandler for dumping and loading the OAuth token.
- **token\_handler\_kwargs** (*dict*, *Optional*) – Keyword arguments to pass to the token\_handler class. These become attributes on the class instance.

### 1.2.1 Properties

MobileClient.**device\_id**

The mobile device ID of the *MobileClient* instance.

MobileClient.**is\_subscribed**

The subscription status of the account linked to the *MobileClient* instance.

MobileClient.**locale**

The locale of the *MobileClient* instance.

Can be changed after instantiation.

ICU locale used to localize some responses. This must be a locale supported by Android.

MobileClient.**tier**

The subscription tier of the *MobileClient* instance.

Can be changed after instantiation.

aa if subscribed, fr if not.

### 1.2.2 Meta

MobileClient.**config**()

Get a listing of mobile client configuration settings.

MobileClient.**device\_deauthorize** (*device*)

Deauthorize a registered device.

**Parameters** *device* (*dict*) – A device dict as returned by *devices*.

MobileClient.**device\_set** (*device*)

Set device used by *MobileClient* instance.

**Parameters** *device* (*dict*) – A device dict as returned by *devices*.

MobileClient.**devices**()

Get a listing of devices registered to the Google Music account.

### 1.2.3 Information

`MobileClient.album(album_id, *, include_description=True, include_songs=True)`

Get information about an album.

**Parameters**

- **album\_id** (*str*) – An album ID. Album IDs start with a ‘B’.
- **include\_description** (*bool*, *Optional*) – Include description of the album in the returned dict.
- **include\_songs** (*bool*, *Optional*) – Include songs from the album in the returned dict. Default: True.

**Returns** Album information.

**Return type** `dict`

`MobileClient.artist(artist_id, *, include_albums=True, num_related_artists=5, num_top_tracks=5)`

Get information about an artist.

**Parameters**

- **artist\_id** (*str*) – An artist ID. Artist IDs start with an ‘A’.
- **include\_albums** (*bool*, *Optional*) – Include albums by the artist in returned dict. Default: True.
- **num\_related\_artists** (*int*, *Optional*) – Include up to given number of related artists in returned dict. Default: 5.
- **num\_top\_tracks** (*int*, *Optional*) – Include up to given number of top tracks in returned dict. Default: 5.

**Returns** Artist information.

**Return type** `dict`

`MobileClient.playlist(playlist_id, *, include_songs=False)`

Get information about a playlist.

**Parameters**

- **playlist\_id** (*str*) – A playlist ID.
- **include\_songs** (*bool*, *Optional*) – Include songs from the playlist in the returned dict. Default: False

**Returns** Playlist information.

**Return type** `dict`

`MobileClient.playlist_song(playlist_song_id)`

Get information about a playlist song.

---

**Note:** This returns the playlist entry information only. For full song metadata, use `song` with the ‘trackId’ field.

---

**Parameters** **playlist\_song\_id** (*str*) – A playlist song ID.

**Returns** Playlist song information.

**Return type** `dict`

`MobileClient.podcast(podcast_series_id, *, max_episodes=50)`

Get information about a podcast series.

**Parameters**

- **podcast\_series\_id** (*str*) – A podcast series ID.
- **max\_episodes** (*int*, *Optional*) – Include up to given number of episodes in returned dict. Default: 50

**Returns** Podcast series information.

**Return type** `dict`

`MobileClient.podcast_episode(podcast_episode_id)`

Get information about a podcast\_episode.

**Parameters** **podcast\_episode\_id** (*str*) – A podcast episode ID.

**Returns** Podcast episode information.

**Return type** `dict`

`MobileClient.song(song_id)`

Get information about a song.

**Parameters** **song\_id** (*str*) – A song ID.

**Returns** Song information.

**Return type** `dict`

`MobileClient.station(station_id, *, num_songs=25, recently_played=None)`

Get information about a station.

**Parameters**

- **station\_id** (*str*) – A station ID. Use 'IFL' for I'm Feeling Lucky.
- **num\_songs** (*int*, *Optional*) – The maximum number of songs to return from the station. Default: 25
- **recently\_played** (*list*, *Optional*) – A list of dicts in the form of {'id': '', 'type'} where *id* is a song ID and *type* is 0 for a library song and 1 for a store song.

**Returns** Station information.

**Return type** `dict`

## 1.2.4 Browse

`MobileClient.browse_podcasts(podcast_genre_id='JZCpodcasttopchartall')`

Get the podcasts for a genre from the Podcasts browse tab.

**Parameters** **podcast\_genre\_id** (*str*, *Optional*) – A podcast genre ID as found in `browse_podcasts_genres`. Default: 'JZCpodcasttopchartall'.

**Returns** Podcast dicts.

**Return type** `list`

`MobileClient.browse_podcasts_genres()`

Get the genres from the Podcasts browse tab dropdown.

**Returns** Genre groups that contain sub groups.

**Return type** `list`

`MobileClient.browse_stations(station_category_id)`

Get the stations for a category from Browse Stations.

**Parameters** `station_category_id` (*str*) – A station category ID as found with `browse_stations_categories`.

**Returns** Station dicts.

**Return type** `list`

`MobileClient.browse_stations_categories()`

Get the categories from Browse Stations.

**Returns** Station categories that can contain subcategories.

**Return type** `list`

## 1.2.5 Explore

`MobileClient.explore_genres(parent_genre_id=None)`

Get a listing of song genres.

**Parameters** `parent_genre_id` (*str*, *Optional*) – A genre ID. If given, a listing of this genre's sub-genres is returned.

**Returns** Genre dicts.

**Return type** `list`

`MobileClient.explore_tabs(*, num_items=100, genre_id=None)`

Get a listing of explore tabs.

**Parameters**

- `num_items` (*int*, *Optional*) – Number of items per tab to return. Default: 100
- `genre_id` (*genre\_id*, *Optional*) – Genre ID from `explore_genres` to explore. Default: None.

**Returns** Explore tabs content.

**Return type** `dict`

## 1.2.6 Listen Now

`MobileClient.listen_now_dismissed_items()`

Get a listing of items dismissed from Listen Now tab.

`MobileClient.listen_now_items()`

Get a listing of Listen Now items.

---

**Note:** This does not include situations; use the `situations` method instead.

---

**Returns** With `albums` and `stations` keys of listen now items.

**Return type** `dict`

`MobileClient.situations(*, tz_offset=None)`

Get a listing of situations.

**Parameters** `tz_offset` (*int*, *Optional*) – A timezone offset from UTC in seconds.

## 1.2.7 Playlists

`MobileClient.playlist(playlist_id, *, include_songs=False)`

Get information about a playlist.

**Parameters**

- **playlist\_id** (*str*) – A playlist ID.
- **include\_songs** (*bool*, *Optional*) – Include songs from the playlist in the returned dict. Default: False

**Returns** Playlist information.

**Return type** *dict*

`MobileClient.playlists(*, include_songs=False)`

Get a listing of library playlists.

**Parameters** **include\_songs** (*bool*, *Optional*) – Include songs in the returned playlist dicts. Default: False.

**Returns** A list of playlist dicts.

**Return type** *list*

`MobileClient.playlists_iter(*, start_token=None, page_size=250)`

Get a paged iterator of library playlists.

**Parameters**

- **start\_token** (*str*) – The token of the page to return. Default: Not sent to get first page.
- **page\_size** (*int*, *Optional*) – The maximum number of results per returned page. Max allowed is 49995. Default: 250

**Yields** *list* – Playlist dicts.

`MobileClient.playlist_create(name, description="", *, public=False, songs=None)`

Create a playlist.

**Parameters**

- **name** (*str*) – Name to give the playlist.
- **description** (*str*) – Description to give the playlist.
- **public** (*bool*, *Optional*) – If True and account has a subscription, make playlist public. Default: False
- **songs** (*list*, *Optional*) – A list of song dicts to add to the playlist.

**Returns** Playlist information.

**Return type** *dict*

`MobileClient.playlist_delete(playlist)`

Delete a playlist.

**Parameters** **playlist** (*dict*) – A playlist dict.

`MobileClient.playlist_edit` (*playlist*, \*, *name=None*, *description=None*, *public=None*)  
Edit playlist(s).

**Parameters**

- **playlist** (*dict*) – A playlist dict.
- **name** (*str*) – Name to give the playlist.
- **description** (*str*, *Optional*) – Description to give the playlist.
- **public** (*bool*, *Optional*) – If True and account has a subscription, make playlist public. Default: False

**Returns** Playlist information.

**Return type** `dict`

`MobileClient.playlist_subscribe` (*playlist*)  
Subscribe to a public playlist.

**Parameters** **playlist** (*dict*) – A public playlist dict.

**Returns** Playlist information.

**Return type** `dict`

`MobileClient.playlist_unsubscribe` (*playlist*)  
Unsubscribe from a public playlist.

**Parameters** **playlist** (*dict*) – A public playlist dict.

`MobileClient.playlist_song` (*playlist\_song\_id*)  
Get information about a playlist song.

---

**Note:** This returns the playlist entry information only. For full song metadata, use `song` with the 'trackId' field.

---

**Parameters** **playlist\_song\_id** (*str*) – A playlist song ID.

**Returns** Playlist song information.

**Return type** `dict`

`MobileClient.playlist_songs` (*playlist*)  
Get a listing of songs from a playlist.

**Parameters:** *playlist* (`dict`): A playlist dict.

**Returns** Playlist song dicts.

**Return type** `list`

`MobileClient.playlist_songs_add` (*songs*, *playlist*, \*, *after=None*, *before=None*, *index=None*, *position=None*)  
Add songs to a playlist.

---

**Note:**

- Provide no optional arguments to add to end.
- Provide playlist song dicts for *after* and/or *before*.

- Provide a zero-based `index`.
- Provide a one-based `position`.

Songs are inserted *at* given index or position. It's also possible to add to the end by using `len(songs)` for `index` or `len(songs) + 1` for `position`.

---

#### Parameters

- **songs** (*dict or list*) – A song dict or a list of song dicts.
- **playlist** (*dict*) – A playlist dict.
- **after** (*dict, Optional*) – A playlist song dict songs will follow.
- **before** (*dict, Optional*) – A playlist song dict songs will precede.
- **index** (*int, Optional*) – The zero-based index position to insert songs.
- **position** (*int, Optional*) – The one-based position to insert songs.

**Returns** Playlist dict including songs.

**Return type** `dict`

`MobileClient.playlist_songs_delete(playlist_songs)`

Delete songs from playlist.

**Parameters** **playlist\_songs** (*dict or list*) – A playlist song dict or a list of playlist song dicts.

**Returns** Playlist dict including songs.

**Return type** `dict`

`MobileClient.playlist_songs_move(playlist_songs, *, after=None, before=None, index=None, position=None)`

Move songs in a playlist.

---

#### Note:

- Provide no optional arguments to move to end.
- Provide playlist song dicts for `after` and/or `before`.
- Provide a zero-based `index`.
- Provide a one-based `position`.

Songs are inserted *at* given index or position. It's also possible to move to the end by using `len(songs)` for `index` or `len(songs) + 1` for `position`.

---

#### Parameters

- **playlist\_songs** (*list*) – A list of playlist song dicts.
- **after** (*dict, Optional*) – A playlist song dict songs will follow.
- **before** (*dict, Optional*) – A playlist song dict songs will precede.
- **index** (*int, Optional*) – The zero-based index position to insert songs.
- **position** (*int, Optional*) – The one-based position to insert songs.

**Returns** Playlist dict including songs.

**Return type** `dict`

`MobileClient.thumbs_up_songs(*, library=True, store=True)`

Get a listing of ‘Thumbs Up’ songs.

**Parameters**

- **library** (*bool, Optional*) – Include ‘Thumbs Up’ songs from library. Default: True
- **generated** (*bool, Optional*) – Include ‘Thumbs Up’ songs from store. Default: True

**Returns** Dicts of ‘Thumbs Up’ songs.

**Return type** `list`

## 1.2.8 Podcasts

`MobileClient.podcast(podcast_series_id, *, max_episodes=50)`

Get information about a podcast series.

**Parameters**

- **podcast\_series\_id** (*str*) – A podcast series ID.
- **max\_episodes** (*int, Optional*) – Include up to given number of episodes in returned dict. Default: 50

**Returns** Podcast series information.

**Return type** `dict`

`MobileClient.podcasts(*, device_id=None)`

Get a listing of subscribed podcast series.

**Parameters:**

**device\_id** (*str, Optional*): A mobile device ID. Default: Use `device_id`.

**Returns** Podcast series dict.

**Return type** `list`

`MobileClient.podcasts_iter(*, device_id=None, page_size=250)`

Get a paged iterator of subscribed podcast series.

**Parameters**

- **device\_id** (*str, Optional*) – A mobile device ID. Default: Use `device_id`.
- **page\_size** (*int, Optional*) – The maximum number of results per returned page. Max allowed is 49995. Default: 250

**Yields** *list* – Podcast series dicts.

`MobileClient.podcast_episode(podcast_episode_id)`

Get information about a podcast\_episode.

**Parameters** **podcast\_episode\_id** (*str*) – A podcast episode ID.

**Returns** Podcast episode information.

**Return type** `dict`

`MobileClient.podcast_episodes(*, device_id=None)`  
 Get a listing of podcast episodes for all subscribed podcasts.

**Parameters:**

**device\_id** (*str*, *Optional*): **A mobile device ID.** Default: Use `device_id` of the *MobileClient* instance.

**Returns** Podcast episode dicts.

**Return type** *list*

`MobileClient.podcast_episodes_iter(*, device_id=None, page_size=250)`  
 Get a paged iterator of podcast episode for all subscribed podcasts.

**Parameters**

- **device\_id** (*str*, *Optional*) – A mobile device ID. Default: Use *device\_id*.
- **page\_size** (*int*, *Optional*) – The maximum number of results per returned page. Max allowed is 49995. Default: 250

**Yields** *list* – Podcast episode dicts.

## 1.2.9 Search

`MobileClient.search(query, *, max_results=100, **kwargs)`  
 Search Google Music and library for content.

**Parameters**

- **query** (*str*) – Search text.
- **max\_results** (*int*, *Optional*) – Maximum number of results per type per location to retrieve. I.e up to 100 Google and 100 library for a total of 200 for the default value. Google only accepts values up to 100. Default: 100
- **kwargs** (*bool*, *Optional*) –

**Any of:**

- 'albums'
- 'artists'
- 'genres'
- 'playlists'
- 'podcasts'
- 'situations'
- 'songs'
- 'stations'
- 'videos'

set to `True` will include that result type in the returned dict. Setting none of them will include all result types in the returned dict.

**Returns**

**A dict of results separated into keys:**

- 'albums'
- 'artists'
- 'genres'
- 'playlists'
- 'podcasts'
- 'situations',
- 'songs'
- 'stations'
- 'videos'

**Return type** `dict`

---

**Note:** Free account search is restricted so may not contain hits for all result types.

---

`MobileClient.search_google(query, *, max_results=100, **kwargs)`  
Search Google Music for content.

#### Parameters

- **query** (`str`) – Search text.
- **max\_results** (`int`, *Optional*) – Maximum number of results per type to retrieve. Google only accepts values up to 100. Default: 100
- **kwargs** (`bool`, *Optional*) –

#### Any of:

- 'albums'
- 'artists'
- 'genres'
- 'playlists'
- 'podcasts'
- 'situations'
- 'songs'
- 'stations'
- 'videos'

set to `True` will include that result type in the returned dict. Setting none of them will include all result types in the returned dict.

#### Returns

**A dict of results separated into keys:**

- albums
- artists
- genres
- playlists

- podcasts
- situations
- songs
- stations
- videos

**Return type** `dict`

---

**Note:** Free account search is restricted so may not contain hits for all result types.

---

`MobileClient.search_library(query, *, max_results=100, **kwargs)`

Search Google Music for content.

#### Parameters

- **query** (`str`) – Search text.
- **max\_results** (`int`, *Optional*) – Maximum number of results per type to retrieve.  
Default: 100
- **kwargs** (`bool`, *Optional*) –

#### Any of:

- 'playlists'
- 'podcasts'
- 'songs'
- 'stations'

set to `True` will include that result type in the returned dict. Setting none of them will include all result types in the returned dict.

#### Returns

**A dict of results separated into keys:**

- 'playlists'
- 'podcasts'
- 'songs'
- 'stations'

**Return type** `dict`

`MobileClient.search_suggestion(query)`

Get search query suggestions for query.

**Parameters** **query** (`str`) – Search text.

**Returns** Suggested query strings.

**Return type** `list`

### 1.2.10 Shuffle/Instant Mix

`MobileClient.shuffle_album(album, *, num_songs=100, only_library=False, recently_played=None)`

Get a listing of album shuffle/mix songs.

#### Parameters

- **album** (*dict*) – An album dict.
- **num\_songs** (*int*, *Optional*) – The maximum number of songs to return from the station. Default: 100
- **only\_library** (*bool*, *Optional*) – Only return content from library. Default: False
- **recently\_played** (*list*, *Optional*) – A list of dicts in the form of {'id': '', 'type': ''} where `id` is a song ID and `type` is 0 for a library song and 1 for a store song.

**Returns** List of album shuffle/mix songs.

**Return type** `list`

`MobileClient.shuffle_artist(artist, *, num_songs=100, only_library=False, recently_played=None, only_artist=False)`

Get a listing of artist shuffle/mix songs.

#### Parameters

- **artist** (*dict*) – An artist dict.
- **num\_songs** (*int*, *Optional*) – The maximum number of songs to return from the station. Default: 100
- **only\_library** (*bool*, *Optional*) – Only return content from library. Default: False
- **recently\_played** (*list*, *Optional*) – A list of dicts in the form of {'id': '', 'type': ''} where `id` is a song ID and `type` is 0 for a library song and 1 for a store song.
- **only\_artist** (*bool*, *Optional*) – If True, only return songs from the artist, else return songs from artist and related artists. Default: False

**Returns** List of artist shuffle/mix songs.

**Return type** `list`

`MobileClient.shuffle_genre(genre, *, num_songs=100, only_library=False, recently_played=None)`

Get a listing of genre shuffle/mix songs.

#### Parameters

- **genre** (*dict*) – A genre dict.
- **num\_songs** (*int*, *Optional*) – The maximum number of songs to return from the station. Default: 100
- **only\_library** (*bool*, *Optional*) – Only return content from library. Default: False
- **recently\_played** (*list*, *Optional*) – A list of dicts in the form of {'id': '', 'type': ''} where `id` is a song ID and `type` is 0 for a library song and 1 for a store song.

**Returns** List of genre shuffle/mix songs.

**Return type** `list`

`MobileClient.shuffle_song(song, *, num_songs=100, only_library=False, recently_played=None)`

Get a listing of song shuffle/mix songs.

**Parameters**

- **song** (*dict*) – A song dict.
- **num\_songs** (*int*, *Optional*) – The maximum number of songs to return from the station. Default: 100
- **only\_library** (*bool*, *Optional*) – Only return content from library. Default: False
- **recently\_played** (*list*, *Optional*) – A list of dicts in the form of {'id': '', 'type'} where id is a song ID and type is 0 for a library song and 1 for a store song.

**Returns** List of artist shuffle/mix songs.

**Return type** *list*

## 1.2.11 Songs

`MobileClient.song(song_id)`

Get information about a song.

**Parameters** **song\_id** (*str*) – A song ID.

**Returns** Song information.

**Return type** *dict*

`MobileClient.songs()`

Get a listing of library songs.

**Returns** Song dicts.

**Return type** *list*

`MobileClient.songs_iter(*, page_size=250)`

Get a paged iterator of library songs.

**Parameters** **page\_size** (*int*, *Optional*) – The maximum number of results per returned page. Max allowed is 49995. Default: 250

**Yields** *list* – Song dicts.

`MobileClient.songs_add(songs)`

Add store songs to your library.

**Parameters** **songs** (*list*) – A store song dict or a list of store song dicts.

**Returns** Songs' library IDs.

**Return type** *list*

`MobileClient.songs_delete(songs)`

Delete songs from library.

**Parameters**

- **songs** (*list*) –
- **library song dict or a list of library song dicts.** (A) –

**Returns** Successfully deleted song IDs.

**Return type** *list*

`MobileClient.songs_play(songs)`

Add play to song play count.

**Parameters** `songs` (*dict* or *list*) – A song dict or a list of song dicts.

**Returns** `True` if successful, `False` if not.

**Return type** `bool`

`MobileClient.songs_rate(songs, rating)`

Rate song.

**Parameters**

- **songs** (*dict* or *list*) – A song dict or a list of song dicts.
- **rating** (*int*) – 0 (not rated), 1 (thumbs down), or 5 (thumbs up).

**Returns** `True` if successful, `False` if not.

**Return type** `bool`

`MobileClient.thumbs_up_songs(*, library=True, store=True)`

Get a listing of ‘Thumbs Up’ songs.

**Parameters**

- **library** (*bool*, *Optional*) – Include ‘Thumbs Up’ songs from library. Default: `True`
- **generated** (*bool*, *Optional*) – Include ‘Thumbs Up’ songs from store. Default: `True`

**Returns** Dicts of ‘Thumbs Up’ songs.

**Return type** `list`

## 1.2.12 Stations

`MobileClient.situations(*, tz_offset=None)`

Get a listing of situations.

**Parameters** `tz_offset` (*int*, *Optional*) – A timezone offset from UTC in seconds.

`MobileClient.station(station_id, *, num_songs=25, recently_played=None)`

Get information about a station.

**Parameters**

- **station\_id** (*str*) – A station ID. Use ‘IFL’ for I’m Feeling Lucky.
- **num\_songs** (*int*, *Optional*) – The maximum number of songs to return from the station. Default: 25
- **recently\_played** (*list*, *Optional*) – A list of dicts in the form of {‘id’: ‘’, ‘type’} where `id` is a song ID and `type` is 0 for a library song and 1 for a store song.

**Returns** Station information.

**Return type** `dict`

`MobileClient.stations(*, generated=True, library=True)`

Get a listing of library stations.

The listing can contain stations added to the library and generated from the library.

**Parameters**

- **generated** (*bool*, *Optional*) – Include generated stations. Default: `True`
- **library** (*bool*, *Optional*) – Include library stations. Default: `True`

**Returns** Station information dicts.

**Return type** `list`

`MobileClient.stations_iter(*, page_size=250)`

Get a paged iterator of library stations.

**Parameters** `page_size` (`int`, *Optional*) – The maximum number of results per returned page. Max allowed is 49995. Default: 250

**Yields** `list` – Station dicts.

`MobileClient.station_feed(*, num_songs=25, num_stations=4)`

Generate stations.

---

**Note:** A Google Music subscription is required.

---

#### Parameters

- `num_songs` (`int`, *Optional*) – The total number of songs to return. Default: 25
- `num_stations` (`int`, *Optional*) – The number of stations to return when no `station_infos` is provided. Default: 5

**Returns** Station information dicts.

**Return type** `list`

`MobileClient.station_songs(station, *, num_songs=25, recently_played=None)`

Get a listing of songs from a station.

#### Parameters

- `station` (`str`) – A station dict.
- `num_songs` (`int`, *Optional*) – The maximum number of songs to return from the station. Default: 25
- `recently_played` (`list`, *Optional*) – A list of dicts in the form of {'id': '', 'type': ''} where `id` is a song ID and `type` is 0 for a library song and 1 for a store song.

**Returns** Station song dicts.

**Return type** `list`

## 1.2.13 Stream

`MobileClient.stream(item, *, device_id=None, quality='hi', session_token=None)`

Get MP3 stream of a podcast episode, library song, station\_song, or store song.

---

**Note:** Streaming requires a `device_id` from a valid, linked mobile device.

---

#### Parameters

- `item` (`str`) – A podcast episode, library song, station\_song, or store song. A Google Music subscription is required to stream store songs.
- `device_id` (`str`, *Optional*) – A mobile device ID. Default: Use `device_id`.

- **quality**(*str*, *Optional*) –

Stream quality is one of:

- 'hi' (320Kbps)
- 'med' (160Kbps)
- 'low' (128Kbps)

Default: 'hi'.

- **session\_token**(*str*, *Optional*) – Session token from a station dict required for unsubscribed users to stream a station song. station['sessionToken'] as returned by *station* only exists for free accounts.

**Returns** An MP3 file.

**Return type** *bytes*

`MobileClient.stream_url(item, *, device_id=None, quality='hi', session_token=None)`

Get a URL to stream a podcast episode, library song, station\_song, or store song.

---

**Note:** Streaming requires a `device_id` from a valid, linked mobile device.

---

#### Parameters

- **item**(*str*) – A podcast episode, library song, station\_song, or store song. A Google Music subscription is required to stream store songs.
- **device\_id**(*str*, *Optional*) – A mobile device ID. Default: Use *device\_id*.
- **quality**(*str*, *Optional*) –

Stream quality is one of:

- 'hi' (320Kbps)
- 'med' (160Kbps)
- 'low' (128Kbps)

Default: 'hi'.

- **session\_token**(*str*) – Session token from a station dict required for unsubscribed users to stream a station song. station['sessionToken'] as returned by *station* only exists for free accounts.

**Returns** A URL to an MP3 file.

**Return type** *str*

### 1.2.14 Top Charts

`MobileClient.top_charts()`

Get a listing of the default top charts.

`MobileClient.top_charts_for_genre(genre_id)`

Get a listing of top charts for a top chart genre.

**Parameters** `genre_id` (*str*) – A top chart genre ID as found with `top_charts_genres`.

`MobileClient.top_charts_genres()`

Get a listing of genres from the browse top charts tab.

## 1.3 MusicManager — Music Manager API client

**class MusicManager** (*username=None, uploader\_id=None, \*, session=None, token=None, token\_handler=<class 'google\_music.token\_handlers.FileTokenHandler'>, token\_handler\_kwargs=None*)

API wrapper class to access Google Music Music Manager functionality.

```
>>> from google_music import MusicManager
>>> mm = MusicManager('username')
```

#### Parameters

- **username** (*str, Optional*) – Your Google Music username. Used to store OAuth tokens for multiple accounts separately.
- **uploader\_id** (*str, Optional*) – A unique uploader ID. Default: MAC address and username used.
- **session** (*GoogleMusicSession, Optional*) – A session compatible with `GoogleMusicSession`.
- **token** (*dict, Optional*) – An OAuth token compatible with `oauthlib`.
- **token\_handler** (*TokenHandler, Optional*) – A token handler class compatible with `TokenHandler` for dumping and loading the OAuth token.
- **token\_handler\_kwargs** (*dict, Optional*) – Keyword arguments to pass to the `token_handler` class. These become attributes on the class instance.

#### **property uploader\_id**

The uploader ID of the `MusicManager` instance.

#### **property uploader\_name**

The uploader name of the `MusicManager` instance.

#### **download** (*song*)

Download a song from a Google Music library.

**Parameters** `song` (*dict*) – A song dict.

**Returns** Song content as bytestring, suggested filename.

**Return type** `tuple`

#### **quota** ()

Get the uploaded track count and allowance.

**Returns** Number of uploaded tracks, number of tracks allowed.

**Return type** `tuple`

**songs** (\*, *uploaded=True, purchased=True*)

Get a listing of Music Library songs.

**Returns** Song dicts.

**Return type** `list`

**songs\_iter** (\*, *continuation\_token=None, export\_type=1*)

Get a paged iterator of Music Library songs.

**Parameters**

- **continuation\_token** (*str, Optional*) – The token of the page to return. Default: Not sent to get first page.
- **export\_type** (*int, Optional*) – The type of tracks to return. 1 for all tracks, 2 for promotional and purchased. Default: 1

**Yields** *list* – Song dicts.

**upload** (song, \*, *album\_art\_path=None, no\_sample=False*)

Upload a song to a Google Music library.

**Parameters**

- **song** (*os.PathLike or str or audio\_metadata.Format*) – The path to an audio file or an instance of `audio_metadata.Format`.
- **album\_art\_path** (*os.PathLike or str, Optional*) – The relative filename or absolute filepath to external album art.
- **no\_sample** (*bool, Optional*) – Don't generate an audio sample from song; send empty audio sample. Default: Create an audio sample using ffmpeg/avconv.

**Returns** A result dict with keys: 'filepath', 'success', 'reason', and 'song\_id' (if successful).

**Return type** `dict`

## 1.4 Sessions — Google Music HTTP Sessions

**class** `GoogleMusicSession` (*client\_id, client\_secret, scope, \*, token=None, \*\*kwargs*)

**request** (*method, url, data=None, headers=None, withhold\_token=False, \*\*kwargs*)

Build and send a request.

Equivalent to:

```
`python request = client.build_request(...) response = client.  
send(request, ...)`
```

See `Client.build_request()`, `Client.send()` and [Merging of configuration][0] for how the various parameters are merged with client-level configuration.

[0]: [/advanced/#merging-of-configuration](#)

## 1.5 Token Handlers — OAuth Token Handlers

```
class TokenHandler (**kwargs)

    abstract dump (token)
        Dump an OAuth token to storage.

    abstract load ()
        Load an OAuth token from storage.

class FileTokenHandler (**kwargs)

    dump (token, *, username=None, client=None)
        Dump an OAuth token to storage.

    load (username=None, client=None)
        Load an OAuth token from storage.
```



## A

album() (*MobileClient method*), 6  
 artist() (*MobileClient method*), 6

## B

browse\_podcasts() (*MobileClient method*), 7  
 browse\_podcasts\_genres() (*MobileClient method*), 7  
 browse\_stations() (*MobileClient method*), 8  
 browse\_stations\_categories() (*MobileClient method*), 8

## C

config() (*MobileClient method*), 5

## D

device\_deauthorize() (*MobileClient method*), 5  
 device\_id (*MobileClient attribute*), 5  
 device\_set() (*MobileClient method*), 5  
 devices() (*MobileClient method*), 5  
 download() (*MusicManager method*), 21  
 dump() (*FileTokenHandler method*), 23  
 dump() (*TokenHandler method*), 23

## E

explore\_genres() (*MobileClient method*), 8  
 explore\_tabs() (*MobileClient method*), 8

## F

FileTokenHandler (class  
     *google\_music.token\_handlers*), 23

## G

GoogleMusicSession (class  
     *google\_music.sessions*), 22

## I

is\_subscribed (*MobileClient attribute*), 5

## L

listen\_now\_dismissed\_items() (*MobileClient method*), 8

listen\_now\_items() (*MobileClient method*), 8  
 load() (*FileTokenHandler method*), 23  
 load() (*TokenHandler method*), 23  
 locale (*MobileClient attribute*), 5

## M

MobileClient (class  
     *google\_music.clients.mobileclient*), 4  
 mobileclient() (*in module google\_music*), 3  
 MusicManager (*class in google\_music*), 21  
 musicmanager() (*in module google\_music*), 4

## P

playlist() (*MobileClient method*), 6, 9  
 playlist\_create() (*MobileClient method*), 9  
 playlist\_delete() (*MobileClient method*), 9  
 playlist\_edit() (*MobileClient method*), 9  
 playlist\_song() (*MobileClient method*), 6, 10  
 playlist\_songs() (*MobileClient method*), 10  
 playlist\_songs\_add() (*MobileClient method*), 10  
 playlist\_songs\_delete() (*MobileClient method*), 11  
 playlist\_songs\_move() (*MobileClient method*), 11  
 playlist\_subscribe() (*MobileClient method*), 10  
 playlist\_unsubscribe() (*MobileClient method*), 10  
 playlists() (*MobileClient method*), 9  
 playlists\_iter() (*MobileClient method*), 9  
 in podcast() (*MobileClient method*), 7, 12  
 podcast\_episode() (*MobileClient method*), 7, 12  
 podcast\_episodes() (*MobileClient method*), 13  
 podcast\_episodes\_iter() (*MobileClient method*), 13  
 in podcasts() (*MobileClient method*), 12  
 podcasts\_iter() (*MobileClient method*), 12

## Q

quota() (*MusicManager method*), 21

## R

request() (*GoogleMusicSession method*), 22

## S

`search()` (*MobileClient method*), 13  
`search_google()` (*MobileClient method*), 14  
`search_library()` (*MobileClient method*), 15  
`search_suggestion()` (*MobileClient method*), 15  
`shuffle_album()` (*MobileClient method*), 16  
`shuffle_artist()` (*MobileClient method*), 16  
`shuffle_genre()` (*MobileClient method*), 16  
`shuffle_song()` (*MobileClient method*), 16  
`situations()` (*MobileClient method*), 8, 18  
`song()` (*MobileClient method*), 7, 17  
`songs()` (*MobileClient method*), 17  
`songs()` (*MusicManager method*), 22  
`songs_add()` (*MobileClient method*), 17  
`songs_delete()` (*MobileClient method*), 17  
`songs_iter()` (*MobileClient method*), 17  
`songs_iter()` (*MusicManager method*), 22  
`songs_play()` (*MobileClient method*), 17  
`songs_rate()` (*MobileClient method*), 18  
`station()` (*MobileClient method*), 7, 18  
`station_feed()` (*MobileClient method*), 19  
`station_songs()` (*MobileClient method*), 19  
`stations()` (*MobileClient method*), 18  
`stations_iter()` (*MobileClient method*), 19  
`stream()` (*MobileClient method*), 19  
`stream_url()` (*MobileClient method*), 20

## T

`thumbs_up_songs()` (*MobileClient method*), 12, 18  
`tier` (*MobileClient attribute*), 5  
`TokenHandler` (class in `google_music.token_handlers`), 23  
`top_charts()` (*MobileClient method*), 21  
`top_charts_for_genre()` (*MobileClient method*), 21  
`top_charts_genres()` (*MobileClient method*), 21

## U

`upload()` (*MusicManager method*), 22  
`uploader_id()` (*MusicManager property*), 21  
`uploader_name()` (*MusicManager property*), 21