

cloudmusic: 网易云爬虫解决方案

一个功能强大的python库，使用最简单易用的方法获取一切你想要的信息。支持python3以上版本

安装

```
pip install cloudmusic
```

简单示例

- 通过歌曲id获取Music对象，并展示部分属性

```
music = cloudmusic.getMusic(1347630432)

print("歌名: {}".format(music.name))
print("歌手: {}".format(music.artist))
print("音频文件url: {}".format(music.url))

# >>>
# 歌名: 白日 日剧《冤罪律师》主题曲
# 歌手: ['King Gnu']
# 音频文件url: http://m10.music.126.net/.../.mp3
```

- 通过歌单id获取Music对象列表，并下载整个歌单的无损品质音频

```
playlist = cloudmusic.getPlaylist(310729011)

for music in playlist:
    music.download(level = "lossless")
```

- 获取一首歌的热评并展示部分信息

```
music = cloudmusic.getMusic(1347630432)

coms = music.getHotComments()

for com in coms:
    print("发布者: {}".format(com['nickName']))
    print("内容: {}".format(com['content']))
    print("获赞数: {}".format(com['likeCount']))
    print("-----")
```

```
# >>>
# 发布者: Akikonknk
# 内容: 特别喜欢日剧的一个原因 大概就是它帮你认清生活的真相之后 依然教会你如何热爱生活
# 获赞数: 19451
# -----
# 发布者: 我藏起来啦嘻嘻
# 内容: 最后一集的犯人真的是应证了那一句“雪崩的时候没有一片雪花是无辜的”
# 获赞数: 6690
# -----
# 发布者: Usio叔叔
# ...
```

- 通过关键字搜索获取前100首歌曲的Music对象，并输出每首歌的评论数量

```
results = cloudmusic.search("白日", 100)

for music in results:
    print(music.getCommentsCount())

# 5745
# 77
# 181
# 560
# 368
# ...
```

参考文档

Music对象

1. Music对象属性

- **url**: 歌曲音频文件链接
- **id**: 歌曲id
- **name**: 歌曲名称
- **rtist**: 歌手名称
- **artistId**: 歌手id
- **album**: 专辑名称
- **albumId**: 专辑id
- **size**: 音频文件大小
- **type**: 音频文件类型（mp3或m4a）
- **level**: 歌曲品质。默认higher
- **picUrl**: 专辑图url

2. Music对象方法

- **download(dirs, level)**

dirs: 可选。下载保持路径。默认为当前文件夹内创建的新的cloudmusic文件夹。
level: 可选，字符型。默认higher。下载品质，有且只有四种选择：standard, higher, exhigh, lossless。

下载歌曲，返回值为下载绝对路径

- **getHotComments(number)**

number: 可选，整型，默认为15。希望获取的评论个数。上限为15个。

获取热评。返回值为列表，列表元素为字典，字典内容：

"likeCount": 获赞数

"content": 评论内容

"time": 评论时间

"userId": 用户id

"nickName": 用户昵称

"avatarUrl": 用户头像url

"vipType": 0表示未开通vip, 10表示开通音乐包, 11表示开通黑胶

"userType": 0表示普通用户, 非0表示特殊用户（明星，丁磊，网红，小秘书等）

- **getComments(number)**

number: 整型，评论个数。数量无限制。

获取最新的评论。返回值与getHotReview相同。

- **getLyrics()**

获取歌词。返回值为列表。元素1为原始歌词，元素2为翻译歌词。

- **getCommentsCount()**

返回值为整型，本首歌的评论数量。

3. 获取Music对象函数

- **getMusic(id/id_list)**

id/id_list: 必须，整型、字符型或列表。歌曲id或由歌曲id组成的列表。

通过歌曲id或一个由id组成的列表生成music对象。当参数为单个id时返回值为单个music对象，当参数为id列表时返回值为music对象组成的列表。

- **getPlaylist(id)**

id: 必须，整型或字符型。歌单id。

通过歌单id生成music对象，参数为歌单id，返回值为歌单内所有歌曲对应music对象组成的列表。

- **search(content, number)**

content: 必须，字符型。搜索关键字。

number: 可选，整形，搜索结果个数，默认为5。

通过关键词搜索获取music对象。返回值为music对象组成的列表。

- **getAlbum(id)**

id: 必须，整型或字符型。专辑id。

和getPlaylist一样。

User对象

1.User对象属性

- **id:** 用户id
- **level:** 用户等级
- **listenSongs:** 累计听歌数量
- **createTime:** 账号创建时间
- **nickName:** 用户昵称
- **avatarUrl:** 头像url
- **city:** 所在城市的行政区划代码
- **province:** 所在省份的行政区划代码
- **vipType:** vip类型。0表示未开通vip，10表示开通音乐包，11表示开通黑胶
- **birthday:** 生日时间戳
- **signature:** 个性签名
- **fans:** 粉丝数量
- **follows:** 关注的用户数量
- **eventCount:** 动态数量
- **playlistCount:** 创建的歌单数量

2.User对象方法

- **getPlaylist()**

获取用户创建和收藏的全部歌单，返回值为一个列表，每个元素是一个歌单信息组成的字典。字典内容：

id: 歌单id;

name: 歌单名称;

creatorId: 创建者id;

createTime: 创建时间;

coverImgUrl: 歌单封面图片url;

updateTime: 歌单最后一次更新时间;

tags: 歌单所属音乐风格，列表形式

- **getRecord(recordType)**

recordType: 可选。默认为0。为0返回所有时间排行，非0返回最近一周排行。

获取用户听歌排行，返回一个列表，元素为字典。字典内容：

score: 分值。100表示听歌次数最多，越小越少；

music: Music对象

3. 获取User对象函数

- **getUser(id)**

id: 必须，用户id。

通过用户id获取User对象，返回值为User对象。

@卡瓦诺猫