Python 爬虫下载喜马拉雅音频文件

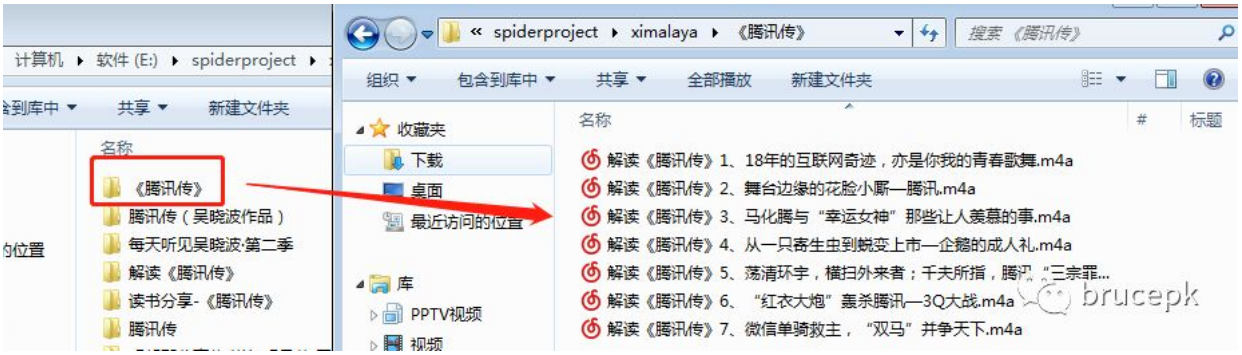
pk哥

阅读文本大概需要 5 分钟

学习，是一个长期的过程。学习的方式也是有很多种的，在家里时间有空闲时间的话可以选择读书，如今在手机上看电子书也方便。pk哥最近看电子书比较多，感觉自己的视力明显下降了。停下来不学习又不行，我想到用听的方式去学习，如今各平台上音频文件还是比较丰富的。大家听得比较多的应该就是喜马拉雅这个平台了。今天我用 Python 把喜马拉雅的音频通过输入关键字查询出来并下载保存在本地。

保存效果

我通过「腾讯传」关键字查询出 6 个音频专辑，以下为其中一个专辑里的 7 个音频文件。



项目环境

语言：Python3

编辑器：Pycharm

程序结构

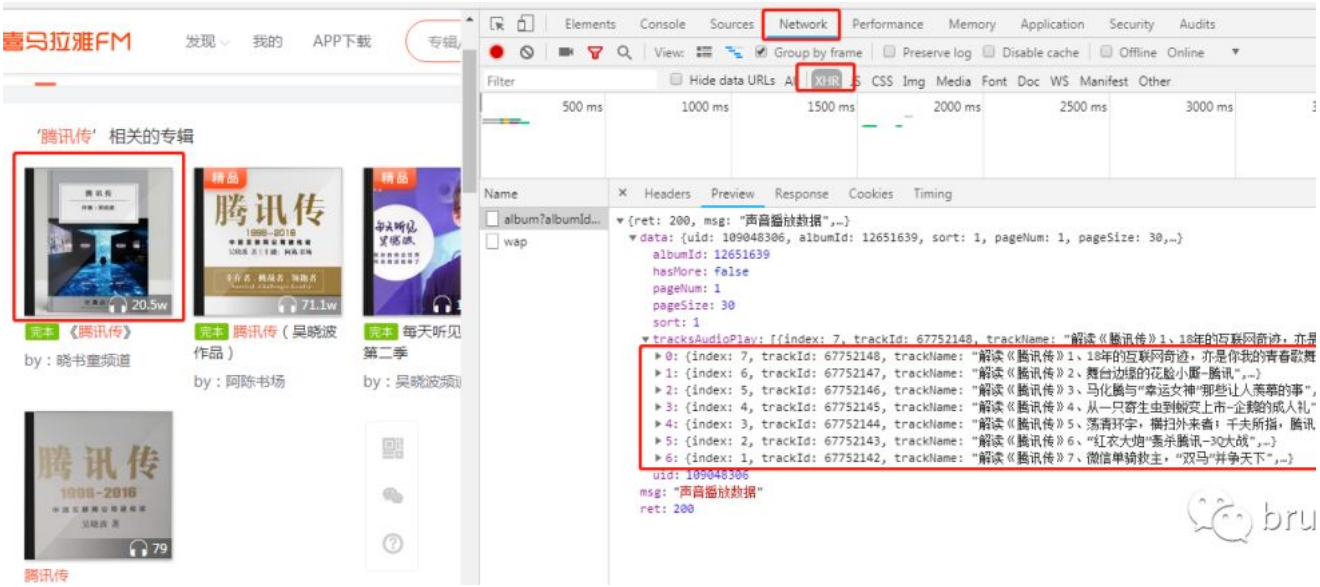
```
import ...

def gethtml(url):...

def getid():...

def downm4a(albumId):...

def mkdir():...

if __name__ == '__main__':
    mkdir()
```


程序主要由四部分组成：

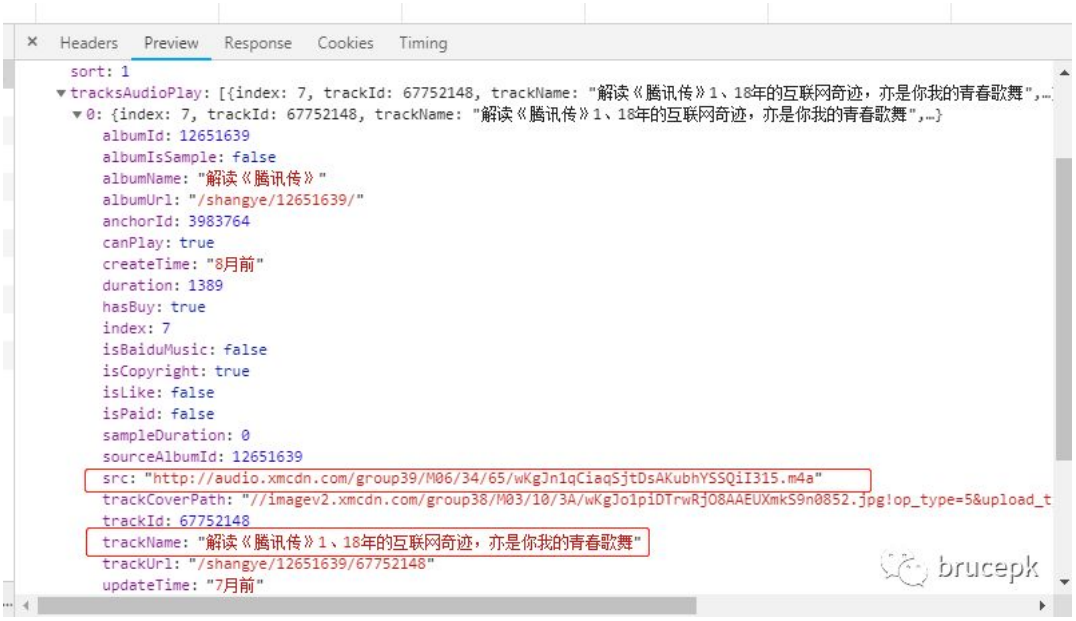
- gethtml()：提取页面 html 信息。
- getid()：获取通过关键字搜索的音频专辑 ID 列表。
- downm4a()：下载对应专辑 ID 下的音频文件。
- mkdir()：把下载的音频保存到相应的文件夹中。

页面分析

我们要下载音频文件，首先我们得要找到下载音频的 url，我们打开浏览器自带的调试工具（我用的是 Chrome），通过快捷键 F12 可快速打开调试工具。调试器切到 Network，我以我最近刚看完的「腾讯传」为例，点击专辑封面中间的播放按钮，该专辑中音频信息中都在 json 格式的数据中。一共有 7 个音频文件。



任意展开一个音频的详细信息，详细信息包括了音频文件的标题和下载链接。找到了音频的下载链接就可以下载音频了，接下来的工作的都围绕怎么获取音频文件的下载链接展开。



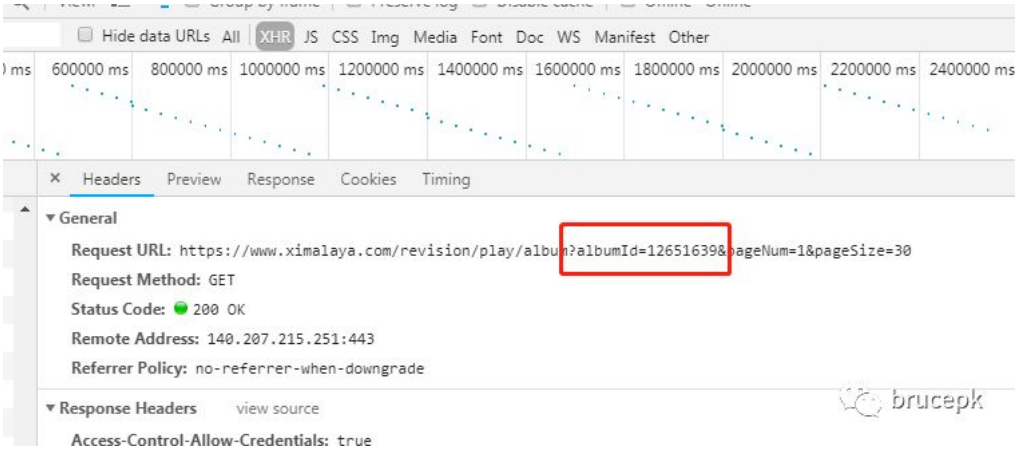
获取页面源码

我们先定义一个获取页面 html 信息的函数。该函数中加入浏览器表头信息 headers，为了安全起见，用的是代理 IP，有兴趣的可以自己做个 IP 代理池，IP 失效后自动替换。

```
def gethtml(url):    # 获取网站 html 信息
    headers = {
        'User-Agent':
        'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safar
    # 用的代理 ip，如果被封的，在http://www.xicidaili.com/换一个
    proxy_addr = {'http': '221.7.255.168:8080'}
    html = requests.get(url, headers=headers, proxies=proxy_addr)    # 请求网页信息
    return html
```

获取专辑信息

接下来我们需要获取专辑的 ID，因为音频的下载链接是通过专辑 ID 拼接的，我们看下刚才包含音频文件名称和下载链接信息的 Headers，可看到专辑链接的组成中 albumId 就是专辑 ID，后面的表示当前页面数和页面最多存放的音频数。



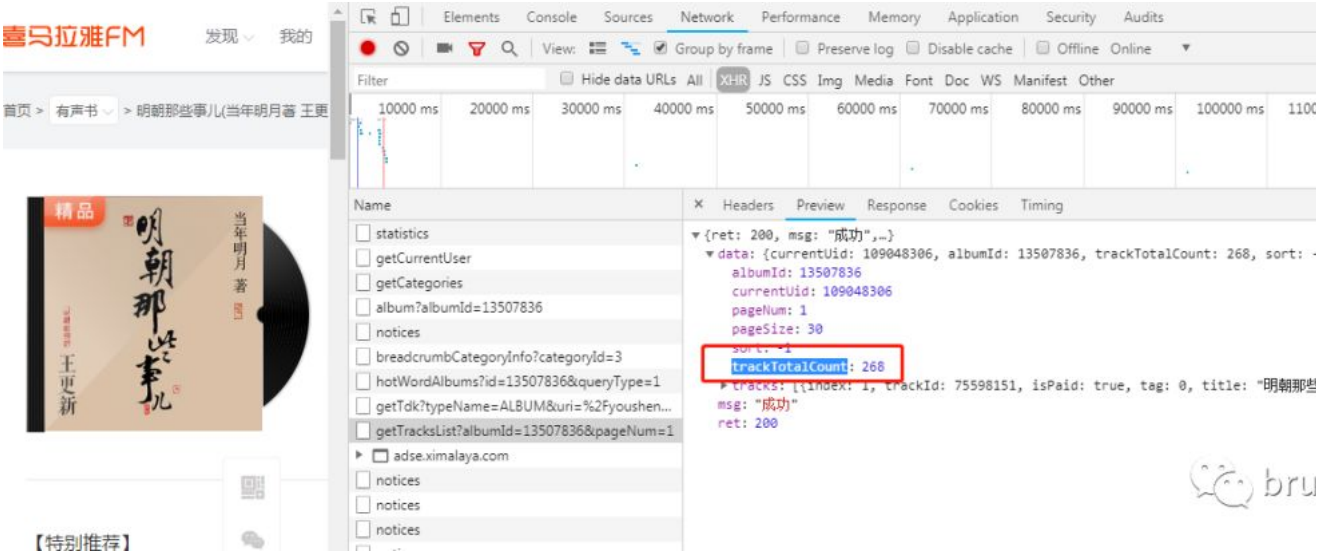
专辑的 ID 信息包含在通过关键字搜索的信息里面。

通过 BeautifulSoup 在页面中提取专辑的 ID 信息，顺便也把专辑标题信息提取出来，当做等下创建专辑目录的名称，主要代码如下。

```
def getid():    # 获取专辑的 id 和标题信息
    keyword = input('请输入你要查找的音频关键字:\n')      # 输入需要下载音频的关键字
    albumurl = 'https://www.ximalaya.com/search/album/{}/sc/p1'.format(keyword)     # 输入关键字，拼接链接
    html = gethtml(albumurl)
    soup = BeautifulSoup(html.text, 'lxml')
    info = soup.select('#searchPage div.search-type div.common-tab-content div.xm-loading ul div '
                       'a.xm-album-title.ellipsis-2')     # 提取音频文件的信息
    idinfo = re.compile('href="/.*?"').findall(str(info))      # 提取专辑中 id
    titleinfo = re.compile('title=".*?"').findall(str(info))      # 提取专辑中标题信息
```


获取页面数

上面的方法获取专辑 ID 信息，接下来我们需要知道专辑下共用多少页的音频文件，我们通过音频总数除以 30 来获取页面数量。音频总数的信息在音频文件列表的 data 里面，下图我用了音频文件数量比较多「明朝那些事儿」举例，一共 268 个音频文件。



有了音频总数，每页的音频数量是 30 个，这样我们就可以算出页面的数量了，分为 3 种情况判断：总数小于或等于 30 个、总数大于 30 个且是 30 的倍数、总数大于 30 个且不是 30 的倍数，相关代码如下。

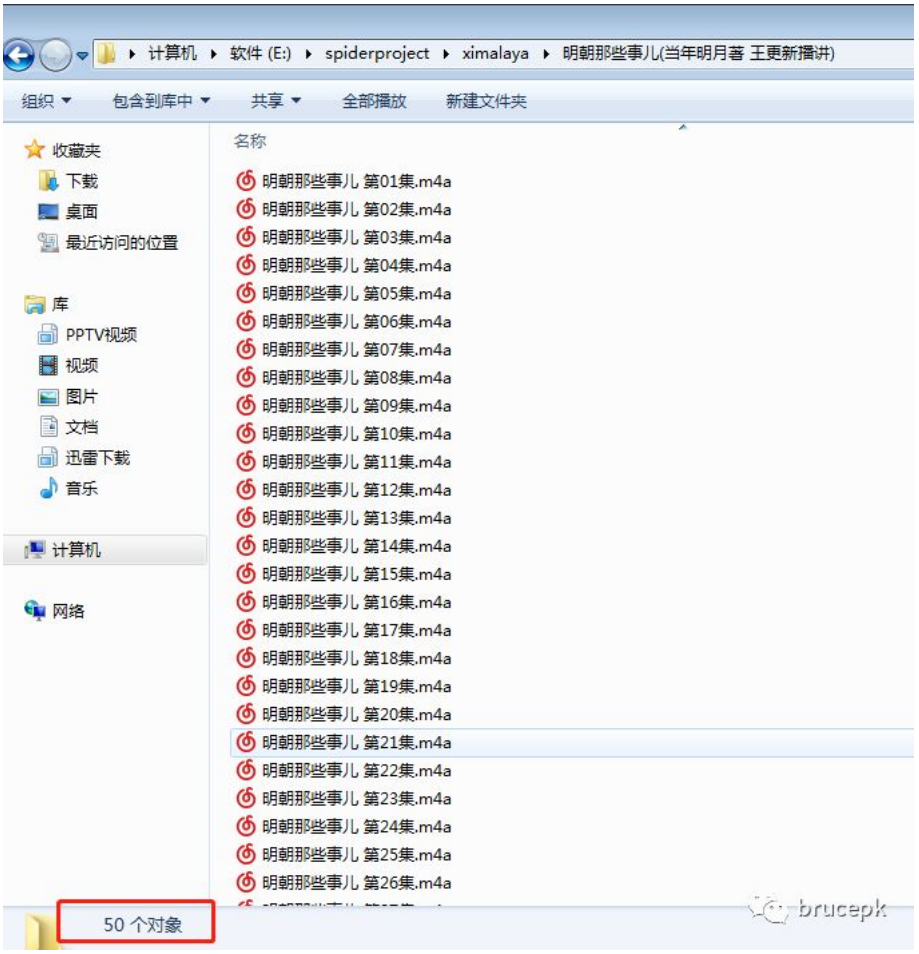
```python
def downm4a(albumId):
    # 获取专辑下的音频总数
    counturl = 'https://www.ximalaya.com/revision/album/getTracksList?albumId={}&pageNum=1'.format(albumId
    chtml = gethtml(counturl)
    cjson = chtml.json()
    trackTotalCount = int(cjson['data']['trackTotalCount'])
    if trackTotalCount < 30 or trackTotalCount == 30:    # 音频数小于等于 30 时，只有一页
        pageNum = 1
    else:
        if trackTotalCount % 30 == 0:                    # 音频数大于 30 时，且是30的倍数时
            pageNum = trackTotalCount // 30
        else:
            pageNum = (trackTotalCount // 30) + 1        # 音频数大于 30 时，不是30的倍数时
```


下载音频文件

专辑 ID、专辑名称、页面数量都有了，接下来就可以下载音频文件了。下载音频时，当音频不足 30 个，需要做下异常处理。当音频文件是付费文件时，无法下载。这时做一个判断，音频的下载链接为 null 或者 None 时，跳出循环去爬取下一个专辑的文件。

```python
    for num in range(1, pageNum+1):
        m4aurl = 'https://www.ximalaya.com/revision/play/album?albumId={}&pageNum={}&pageSize=30'.format(albu
        mhtml = gethtml(m4aurl)
        mjson = mhtml.json()
        for i in range(30):      # 一个页面最多30个音频文件
            try:
                trackName = mjson['data']['tracksAudioPlay'][i]['trackName']      # 提取音频标题
                src = mjson['data']['tracksAudioPlay'][i]['src']                  # 提取可下载链接
                print(trackName)
                print(src)
                if str(src) in('null', 'None'):           # 如果为付费音频，则跳出循环，继续下载下一个专辑
                    print('此为付费音频，无法下载')
                    break
                data = requests.get(src).content
                with open('%s.m4a' % trackName, 'wb') as f:       # 下载音频
                    f.write(data)
            except IndexError:
                print('当前专辑已爬取完成！')
                continue
```


音频的下载链接为 null 或者 None 的情况，这里以「明朝那些事儿」为例，通过「明朝那些事儿」关键字爬取的其中一个专辑的音频文件，总共只爬取了 50 个，后面的音频文件都没有提供下载链接，所以无法下载。



建立目录存放音频

为了让下载下来的音频文件有序的存放在以专辑名称命名的文件夹下，我们用代码自动创建目录并把对应文件下载到该目录下。

```
def mkdir():    # 判断目录是否存在，不存在的话则自动创建
    ids, titles = getid()
    for title, albumId in zip(titles, ids):
        print(title)
        path = 'E:\\spiderproject\\ximalaya\\{}'.format(title)    # 以音频名称命名
        isExists = os.path.exists(path)
        if not isExists:
            print('创建目录{}'.format(title))      # 目录不存在则创建一个
            os.makedirs(path)                      # 创建目录
            os.chdir(path)                         # 切换到创建的文件夹
            downm4a(albumId)                       # 调用函数下载音频到该目录下
        else:
            print('{}目录已存在,即将保存！'.format(title))
            os.chdir(path)                         # 切换到创建的文件夹
            downm4a(albumId)                       # 目录已存在时直接保存
        time.sleep(int(format(random.randint(2, 6))))     # 随机等待
```


后记

本文的目的是把喜马拉雅上免费的音频下载到本地，传到手机里，方面大家保护视力的情况下随时都可以学习。当然，流量充足的也可以在 APP 上在线听。

此项目是通过输入关键字去搜索音频专辑下载的，对于有些关键字没有对应音频的情况下，系统会把推荐音频给你，所以为了提高大家的效率，大家运行代码前，先在喜马拉雅网站输入你需要搜索的关键字，看是否有相关的音频，有的话再运行代码。一般热度比较高的音频专辑都比较靠前，下载了自己需要的音频专辑后，如果后面的专辑不需要停止运行代码即可。

源码可在公众号回复「听」获取。

推荐阅读：

人人公众号时代，如何挑选优质公众号？

实时 12306 车票查询

人必有痴，而后有成

文章转载自公众号

brucepk

brucepk