写个抓取网易云音乐精彩评论的爬虫

董伟明 · 12 天前

标题有点长啊。

其实写爬虫是一个很微小的事情， 在12年 和14年 ，我却单靠这2个爬虫获得了offer，所以爬虫真的算Python工程师的必修课。这2个爬虫其实都很cute，现在看起来有很多地方其实理解的不够深入，也有一些实现地方做的不好，但是毕竟是当时的我，就保留着吧。

其实拿到数据怎么用，比如做数据分析，做创业项目原始启动数据，数据可视化等等。那我就利用Web开发的优势，把数据在页面上展示出来吧。无图无真相，先上图（知乎不支持并排图片，哎）：

按评论数

初学者
薛之谦 ——首一首来吧...《初学者》 词曲 薛之谦 感谢你的聆听...

悟空
944黄小刀 –今天去看了大圣归来。我旁边有个小孩儿问他妈妈 "这个不是动画片么？为什么有这么多大人来看？" 他妈妈回答： "因为他们一直在等大圣归来啊，等啊等啊，就长大了." ——泪目~

演员
一加二等于宝晴标 –实在不明白为什么薛之谦不红，现在华语乐坛的新人都是些什么妖魔鬼怪

See You Again
KingFFFFZH –就在保罗沃克去世前13天，我高中最好的一个兄弟也因为意外去世了。曾经一起在寝室看了速度1，之后一起看了速度5速度6。曾经也同样举爱速度与激情。今天去看了第七部，保罗沃克不在了。我兄弟也不在了。当听到这首see you again真的眼眶湿润了。献给保罗沃克，献给我的兄...

晴天
蛋蛋是圆D –高一听的，那时候遇到了孩儿他妈，然后就这么幸福下来了

最佳歌手
许嵩 –感谢每一位前来聆听的朋友。你们最佳[爱心]

演员
pooocky ——个人能有多不正经，就能有多深情。

We Are Young
帕瓦罗蒂 –主唱长得真丑

We Are Young
速写素描 –我跟主唱长得很像！！！

悟空
Victoria-O –有个外国人问我为什么喜欢孙悟空。我回答他:超人。钢铁侠，美国队长为你们维护正义七八十年。而孙悟空，为我们斩妖除魔，五百年。你们有很多英雄。我们只有他一个。

The Phoenix
Amazingsong –妈的，开车听感觉自己在开坦克...

Intro: The Dawn
你吃冰棒我吃棍 –9年前，魔兽世界公测时我有女朋友，现在，魔兽世界还在电脑里，陪伴我9年的女朋友却没了

Booty Music
爱听歌的灰灰菇凉 –我的帝帝，之前没听懂歌词，推荐给暗恋两年的男生听了。不知道听听后怎么想[流泪][流泪][流泪][流泪][流泪]

绅士
安格其 ——个人能有多不正经就能有多深情。

大鱼
动不动就饿0 –真的觉得中国人很奇怪！[嘲讽]对自己国家的文化没有任何自信 为什么好不容易中国有了不错的作品 国人就开始诋毁 说什么抄袭千与千

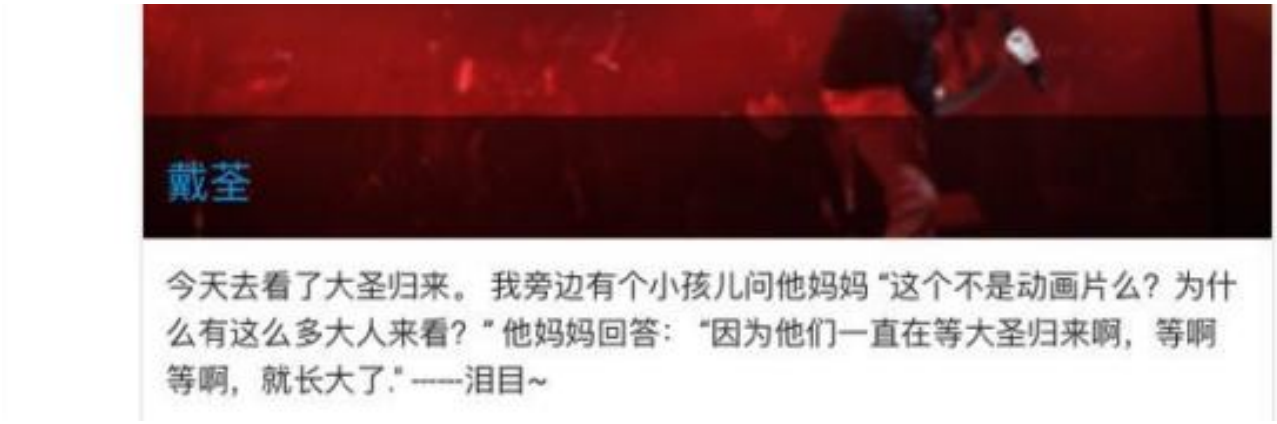
We Are Young


速写素描--我跟主唱长得很像！！！
悟空
Victoria-O --有个外国人问我为什么喜欢孙悟空。我回答他:超人，钢铁侠，美国队长为你们维护正义七八十年。而孙悟空，为我们斩妖除魔，五百年。你们有很多英雄。我们只有他一个。
The Phoenix
Amazingsong --妈的，开车听感觉自己在开坦克...
Intro: The Dawn
你吃冰糕我吃棍 --9年前，魔兽世界公测时我有女朋友，现在，魔兽世界还在电脑里，陪伴我9年的女朋友却没了
Booty Music
爱听歌的灰灰菇凉 --我的乖乖，之前没听懂歌词，推荐给暗恋两年的男生听了。不知道听听后怎么想[流泪][流泪][流泪][流泪][流泪]
绅士
安略其 --一个人能有多不正经就能有多深情。
大鱼
动不动就饿0 --真的觉得中国人很奇怪！[嘴嘴]对自己国家的文化没有任何自信 为什么好不容易中国有了不错的作品 国人就开始诋毁 说什么抄袭千与千寻，但是大鱼的故事 还有人物的背景 和千与千寻千差万别 而且都是中国自己的东西 客家土楼 山海经 凤凰 龙 还有逍遥游 这都是满满的中国风！
南山南
费夕 --"你在南方的艳阳里大雪纷飞，我在北方的寒夜里四季如春"，这首歌深刻地点明了北方有暖气南方无暖气所导致的人民矛盾。
Counting Stars

IOS访问是这样的：


●●●●○ 中国移动
下午3:06
35%
192.168.1.242


按评论数
薛之谦
薛之谦
一首一首来吧... 《初学者》 词曲 薛之谦 感谢你的聆听....
944黄小刀



PS: 移动端图片有点糊是因为我使用了小尺寸，担心太浪费用户手机流量。而且在Web端也是可以自由的切换这2种展示模式。

我在用随机刷着玩的时候，看到了这么一条：

```
AJAPKK：阿黛尔：你经历过绝望吗
```

我就点进去听了下这首廖佳琳 版本的Rolling in the deep ，额那一天我单曲循环了一下午的这首神曲。建议一边听一边继续向下看。

其次是发现最热的评论中薛之谦的歌曲占了好几个，好吧我得先承认，之前认为喜欢参与综艺节目的歌手歌唱的都不行，尤其薛之谦以段子手而著名。但是看到总榜之后，我还是挨个听了他的歌，觉得其实还行。

在知乎，感觉没用过Python写爬虫都不好意思和人打招呼。我想写篇爬虫的文章，所以就开始找需求，其实一开始我是准备爬豆瓣害羞组（不懂得可以搜一搜），但是连续2个深夜2点去蹲守，发现现在那几个小组不够劲爆，而且量也太少，而且担心发了文章有人举报我 ~=(๑•́ ₃ •̀๑) ，所以作罢。

上班的路上，除了看kindle，我也经常会带着耳机听网易云音乐（简称网云吧）里面收藏的歌，额，其实经常还能看到好多好玩的评论的，有辣眼睛的，有悲伤的，有总结很精辟的，有讲一些不是同年代人不会懂的。可以先预览下 网易云音乐有哪些有趣的评论？ 和 网易云音乐评论量最高的歌曲有哪些？在这些评论里你又发现了什么值得品味的故事？

等不及了？ 它们都在这里 云音乐评论 ， 里面有 28925 个歌手（组合）演唱的 710182 首歌曲中的 720699 条精彩评论。

文章的头图下面那10个，基本就是被最多评论的歌曲的专辑前十了。

这个项目地址是： GitHub - dongweiming/commentbox （fork时别忘记点赞哦😗），使用的技术：

1. 后端： Flask + Mongoengine + Mako + requests + Redis + lxml + concurrent.futures

2. 前端：React + Mobx + Fetch + Material-UI + ES6 + Webpack + Babel

今天先和大家聊聊写个爬虫需要熟悉哪些知识，思路是什么，怎么实践的，欢迎关注专栏，节后我再聊后端和前端的实现， 也有使用Flask的经验。

需求分析

既然要爬整站的热门评论，就要找到「**入口**」。什么是入口呢？就是类似聚合页，比如抓知乎的全部问题和答案，我的思路是先爬 话题广场 ， 爬每个一级话题，再去爬话题下子话题，如 教育 - 热门问答 。 然后不断翻页就好了。由于一个问题可以有多个话题标签，需要注意缓存已爬取和正在爬取的答案页面地址，防止重复爬取。

那网云呢？评论在歌曲下，歌曲在歌手下。找到全部歌手就好了。所以先爬 music.163.com/#... ， 找到规律，按照歌手所在地区和首字母翻就能遍历了。

其次是预估最后的结果量。其实我这几十万的只是网云评论数据的一个小小小子集，主要是看爬取要花的时间，以及可提供存储的空间。对这个需求来说，有些歌曲都是几十万个评论，我用一台非闲置的服务器抓取，肯定一年也抓不完，**不要忘记，抓取的瓶颈不在你使用多线程，多进程或者asyncio，主要在对方对你的抓取的容忍程度以及在爬和反爬策略伤的博弈的结果。**而且我的VPS是1G内存，考虑爬取下来存入MongoDB的空间，并要给Redis预留缓存使用内存的使用量，所以我决定：

1. 每个歌手只抓取Ta最热门的50首歌曲。

2. 每首歌只要最热的前10条评论。

当时目测歌手数量在1-2w，而有些冷门歌手没有50首歌曲，或者热门评论不足，也就是大概200万条左右（理想情况下 1-2w * 50 * 10）。实际上和我预期的少了不少，但是还是让我的1G VPS捉襟见肘了... #论确定需求的重要性#

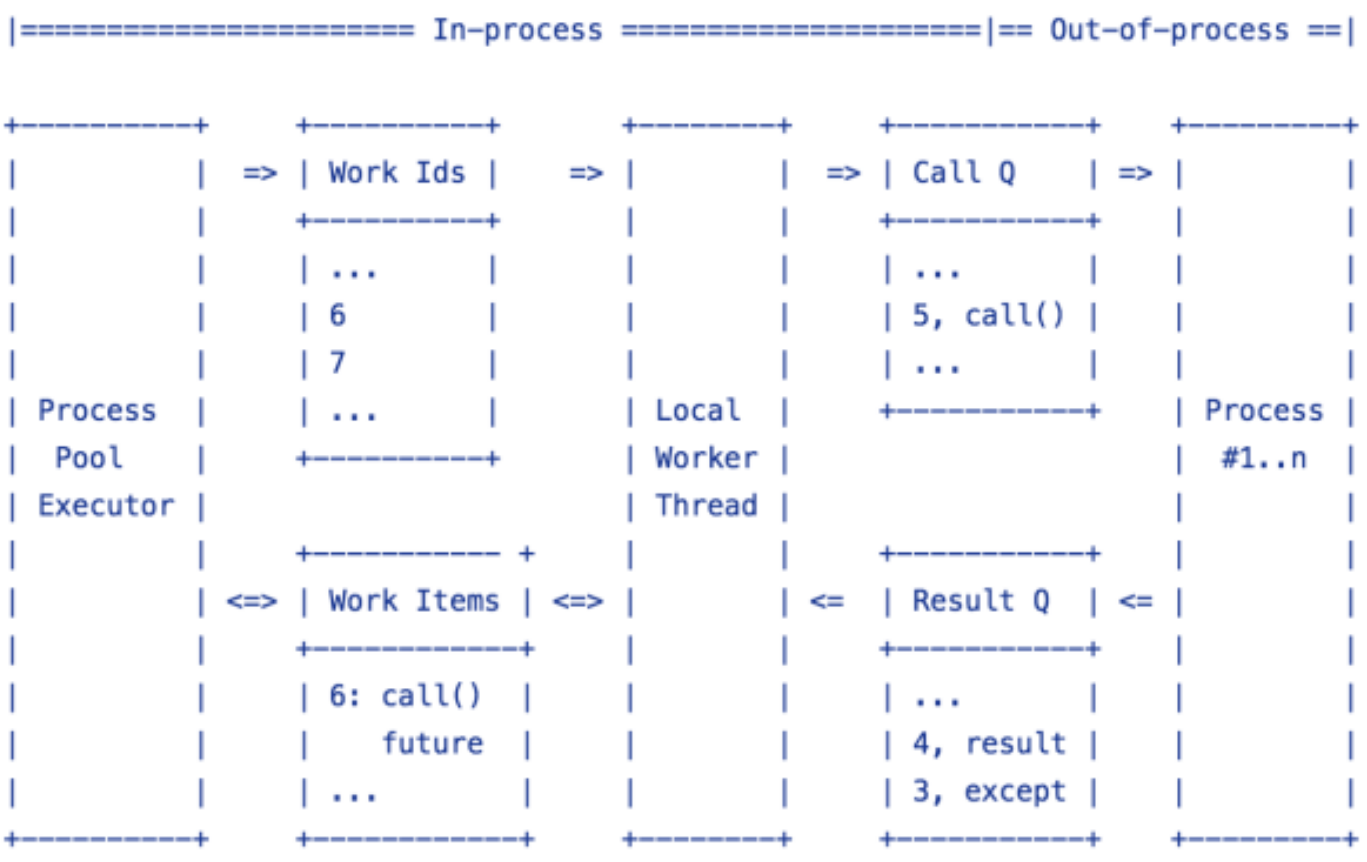
技术选型

我使用过各种解析页面的库，现在一般只使用BeautifulSoup（bs4）或者lxml，如果页面比较

简单，标签写的比较严谨且需求单一或者一次性一般都用BeautifulSoup，比如豆瓣；复杂的、未来会一直都在用的选择lxml，比如淘宝这种页面被各业务线拼的模板。

我其实推荐大家好好学习xpath，这也是我选择lxml的一个原因。那为什么要用xpath，假如你只是爬一个站，其实无所谓，假如你要爬各种同类型的网站，比如豆瓣东西的发布东西 ，它支持发布数十个网站，肯定得让抓取框架化，因为发布东西要的那些字段都是定的，比如标题，价格，商品图片，用xpath新人可以不关心它怎么运作的，只是按照xpath寻找对应的元素去获取之，几分钟就能学会如何对一个新的网站进行支持，假如使用BeautifulSoup，你要写一坨坨的解析、遍历。而xpath永远都是一句搞定。

接着说 concurrent.futures， 这是一个在Python 3.2 的时候就被放进标准库的模块，它高度抽象出了异步执行任务的接口，把队列的使用隐藏起来，而且多进程和多线程接口统一，对于使用来说，切换多进程和多线程很简单。这比你写一大坨的多进程或者多线程的代码要简单很多。它的数据流 是这样的：



我们这个需求中抓取逻辑中，这样使用：

```
from concurrent.futures import ProcessPoolExecutor
```

```
with ProcessPoolExecutor(max_workers=2) as executor:
    for artist_id in unprocess_artist_list():
        executor.submit(parser_artist, artist_id)
```

可以把它理解成一个2个进程的进程池。如果你的服务器CPU个数更多，处理能力更强，不要吝啬加大这个值哦。

最后说requests。这个太有名，不用它的人可能不理解为啥都用它，它的描述是「Python HTTP Requests for Humans」，是的，其实并没有人强制要用它，而且不用它也是可以的，只是你得自己写一大坨的代码才能支持会话，Cookie，代理等需求。对于没有自虐倾向的人来说，Python标准库提供的方案确实太底层了。我之前还特意研究了下为了这么好的东西不直接放进标准库？看 Consider Requests' Inclusion in Python 3.5's Standard Library · Issue #2424 其中多个大神出没哦。

怎么样不被发现

1. 不要用一个IP狂爬。所以要准备一堆可用的代理IP，如果公司有额外的比较闲的IP最好了，闲着也是闲着，在不影响正常业务的提前下，多换IP。否则就要想办法获取免费代理。我的书中这个地方有写。

2. **勤换UA。**我看很多人喜欢在配置中列一些UA, 其实吧，可以使用 GitHub - hellysmile/fake-useragent: up to date simple useragent faker with real world database 。其实我也推荐大家伪装成各大搜索网站的UA， 比如Google UA 有这样一些 Google 抓取工具 ，说到这里，有的网站，你添加referfer字段是搜索网站也是有用的，因为网站是希望被索引的，所以会放宽搜索引擎的爬取策略。

3. **爬取间隔自适应。**就是已经限制了你这个IP的抓取，就不要傻傻重复试，怎么滴也得休息一会。网易云音乐操作起来比较简单，sleep一下就好了。其实sleep的间隔应该按情况累加，比如第一次sleep 10秒，发现还是被约束。那么久sleep 20秒... 这个间隔的设置已经自适应的最终效果是经验值。

4. **验证码识别。**现在攻防让验证码技术层出不穷，其实好多都是自己写算法识别，并不开源，开源的就是tesseract，还可以借用百度识图平台 试试。我个人还是倾其所有的做好其他的地方，不要让人家弹出验证码让我输入。

开始爬

首先一定要防止「由于异常等原因造成爬虫程序错误，重新启动还会重新爬」的尴尬。我建了一张Process表，用来存爬取的状态：开始爬取置状态为「PENDING」，抓取完成置状态为「SUCCEEDED」（当然也有失败，比如页面解析未覆盖到情况造成的失败，之后失败的状态应该没有条目才对，否则就要去兼容）。每次抓取程序启动都会检查哪些PENDING的先抓完，抓过的直接忽略去下一个。

真的数据Model包含4个：Artist（歌手）、Song（歌曲）、Comment（评论）和User（评论人），我们感受一下抓取的过程（截取重要部分）：

```
def parser_artist(artist_id):
    create_app()  # Flask应用要先初始化
    process = Process.get_or_create(id=artist_id)  # Process以歌手为单位
    if process.is_success:  # 如果已经成功直接返回了
        return

    tree = get_tree(ARTIST_URL.format(artist_id))  # 使用requests获取页面

    artist = Artist.objects.filter(id=artist_id)
    if not artist:  # 如果之前没抓过
        artist_name = tree.xpath('//h2[@id="artist-name"]/text()')[0]
        picture = tree.xpath(
            '//div[contains(@class, "n-artist")]//img/@src')[0]
        artist = Artist(id=artist_id, name=artist_name, picture=picture
        artist.save()
    else:  # 如果之前抓过，但是该歌手的歌曲没抓完
        artist = artist[0]
    song_items = tree.xpath('//div[@id="artist-top50"]//ul/li/a/@href')
    songs = []
    for item in song_items:
        song_id = item.split('=')[1]
        song = parser_song(song_id, artist) # 进入抓取和解析歌手模式
        if song is not None:
            songs.append(song)
    artist.songs = songs
    artist.save()
    process.make_succeed()  # 标记歌手下的热门歌曲的热门评论抓完
```

的，思路可见网易云音乐新版WebAPI分析。

无耻的广告：《Python Web开发实战》上市了！

正文完

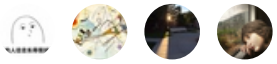
欢迎加入QQ群522012167，或者WEB上扫码进QQ群：

群名称：Python学习2群
群　号：522012167

「打赏越多，更新越快」

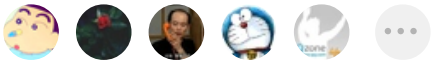
赞赏

4 人赞赏



Python　爬虫（计算机网络）　Flask

591 分享 举报



文章被以下专栏收录

Python之美

发现Python之美，主要针对Web开发、Python进阶等

进入专栏

30 条评论

写下你的评论

张大炮

mark

11 天前

呵呵

少local_settings.py文件？？？？？？？？

11 天前

西伯利亚寒流

MARK!

10 天前

刘同春

精彩

10 天前

苏好铁

书已入手。数据流的图是用emacs画的？

7 天前

本根

学习学习

5 天前

shylock 回复 **爆炸喵** 查看对话

我今天刚取消订单了，然后去z.cn 上买。我都等了一个月了(9.6预购下的订单)

2 天前

爆炸喵 回复 **shylock** 查看对话

你发货了告诉我。我也取消订单去换，我已经点草某东客服几次了每次都跟我说一个星期内发货

2 天前

shylock 回复 **爆炸喵** 查看对话

昨天亚马逊下单，当天就到了。

1 天前

爆炸喵 回复 **shylock** 查看对话

看到你以后我马上取消了狗东的订单跑去亚马逊下单了

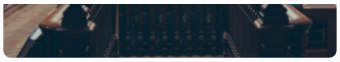
1 天前

上一页 1 2 3

推荐阅读

Flask最佳实践

这是 写个抓取网易云音乐精彩评论的爬虫 - Python之美 - 知乎专栏 的续篇。本节将主要分享 GitHub - dongweiming/commentbox: 网易云音



董伟明 · 4 天前 发表于 Python之美

我的Python订阅列表

除了经常看Github Trending，我还会通过订阅列表等渠道看一些程序员做的有意思的事和分享。之前这个专栏聊过「Python书籍推荐」和「教你阅读P… 查看全文 >

董伟明 · 24 天前 发表于 Python之美

做鸭食谱 / 做这道芋艿烧鸭，要三思

周一收了一只嫩鸭，周二收了一袋新鲜的芋艿。我一般遇到大只的家禽，都会丢进高压锅里焖成高汤，是最方便的晚餐选择。但是嫩鸭烧成汤，味道总… 查看全文 >

SansanL · 19 天前 · 编辑精选 发表于 Painsan · 麵包少女

咖啡酸是什么鬼？有人爱之，有人恶之

这里关闭评论，删除所有评论，没心思也没时间陪撕，但凡个人见解，总有出入，你这么拽，这么十全十美，你怎么不上天呢，在我这里瞎逼逼。不看… 查看全文 >

Alice大鲸鱼 · 2 个月前 · 编辑精选 发表于 Prague coffee