《中国有嘻哈》这款综艺带火了中国的嘻哈音乐，大家问好也都变成了：你有freestyle吗？

大家都看到了这篇高大上的微信推送文章了吧。

没看到也不要紧，传送带在这里-->爱票子也爱妹子：300万字歌词分析看中国rapper到底在唱什么(https://mp.weixin.qq.com/s?timestamp=1502886261&src=3&ver=1&signature=An906sUaINR83Xfe*OGB3*LKxqCATr92XmtqVzGZN*NOL:

真心觉得寒小阳老师的数据分析技术很厉害~还有小编的文笔也很赞~

我主要负责了数据采集的部分。通俗点就是编写一个爬虫，把大量歌曲歌词爬下来。

确定爬取目标

网易云音乐中国嘻哈榜-主播电台 (http://music.163.com/#/djradio?id=169) 吸引了我们的注意。这是最贴近《中国有嘻哈》这个热点的。随着综艺的播出，每周都有新的节目歌曲更新。现在已经到372期了。

首先可以看到节目列表。

点击“节目”进入另一个页面：歌曲列表。发现这期节目有10首歌。

节目包含歌曲列表（10首歌）　　收起

1	小人物	04:28	D-MIX	中国嘻哈榜
2	红眼	04:03	元帅/依兴驰	中国嘻哈榜
3	阎王（Beats By TYRX）	03:05	DM	中国嘻哈榜
4	霍元甲(Prod By.ATYANG)	03:50	JarStick	中国嘻哈榜
5	水浅王八多（prod by 朴冉）	03:26	王天放Fra...	中国嘻哈榜
6	玻尿酸	04:04	J-Sleeper...	中国嘻哈榜
7	Real Life	03:35	孔令奇/满...	中国嘻哈榜
8	G.O.D	03:53	畸形儿-De...	中国嘻哈榜
9	EMP	04:24	低调组合	中国嘻哈榜
10	Each·I	03:26	Big Daddy...	中国嘻哈榜

点击“歌曲”进入歌曲主页，发现了我们想要的歌词，眼前一亮~

生成外链播放器

单曲 • 小人物

歌手：D-MIX

所属专辑：中国嘻哈榜

播放 + 收藏 分享 下载 (240)

从南到北再从东部到西部
打遍天下我带着黄色的皮肤
Shut up f**ka
说中文的小人物
Shut up f**ka
英雄从不问出处
从南到北再从东部到西部
打遍天下我带着黄色的皮肤
Shut up f**ka
说中文的小人物
Shut up f**ka
英雄从不问出处
就像星爷说的

展开

“展开”这个刺眼的js标签让歌词内容部分遮掩。

我们不得不来个selenium模拟鼠标点击，才能得到想要的歌词吗？

其实不必这样，有一个简单的方法：

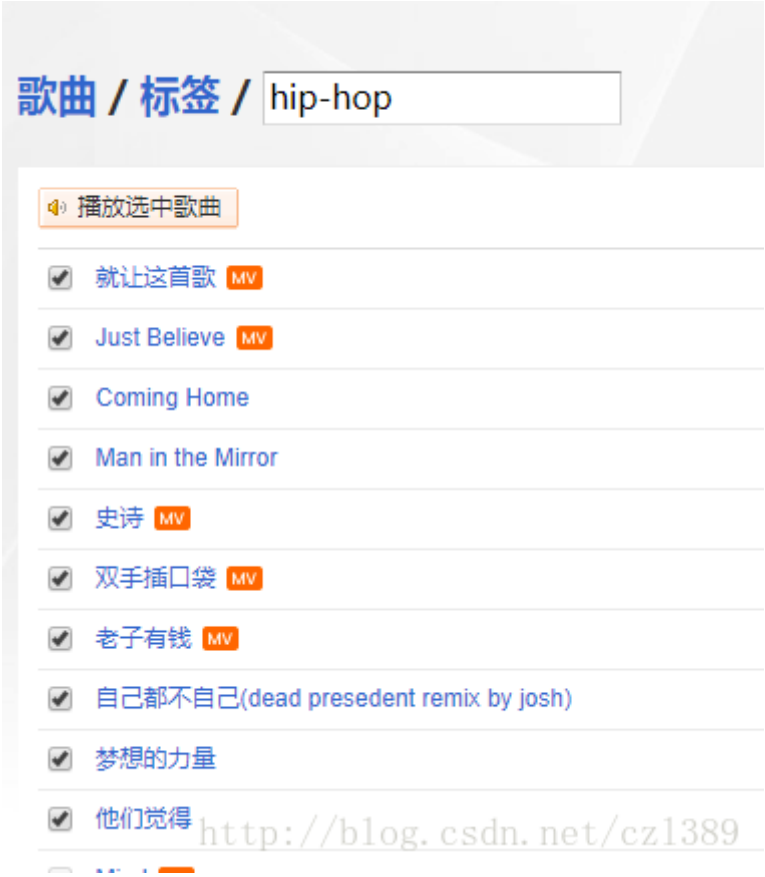
> 取得节目列表下所有歌曲信息
>
> 提示：歌词 -> 使用 http://music.163.com/api/song/media?id=421203370(歌曲编号)来取得。(http://music.163.com/api/song/media?id=421203370(歌曲编号)来取得。)

当时电台也已经更新了208期了，如果每期有10首的，将会得到约2000首歌。

后来，将全部数据爬下来才证明是too naive 了。歌曲重复实在是太多，去重之后，发现才368首，368首！

痛定思痛，我们把目光瞄向了虾米音乐。在搜索框中输下"hip-hop",

回车，我们发现虾米音乐可以搜索特定标签的歌曲，就比如hip-hop。

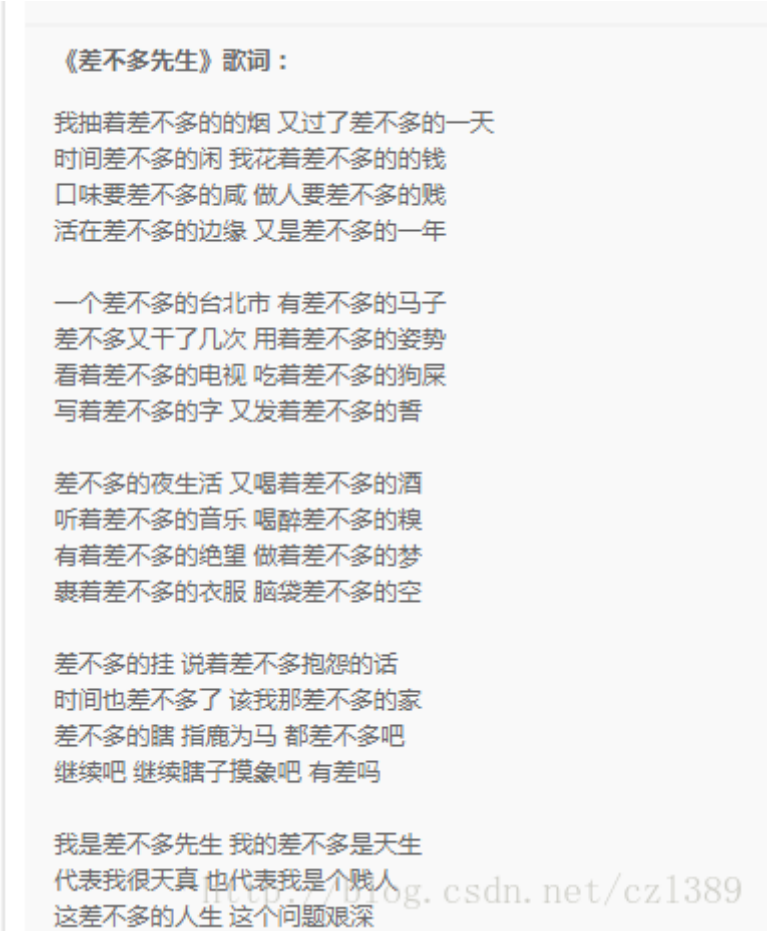


往后翻页，总共有401页。而且url很有规律，可以直接访问。这个好。

偷偷估算了一下，401页，每页30首，可以得到1万两千首，amazing！

不过这次，首先爬取了url,并进行筛重。最后,剩余能有五六千首的样子。

它---虾米，歌词的页面。



好了，爬取目标已确定。

编写爬虫

看一下代码。

使用的是scrapy爬虫框架，很灵活很强大。

与正则表达式、beautifulsoup库相比，有很方便的自定义配置和定制。

在这个框架中，工作主要集中在

- items
- spiders

可能还会用到

- pipelines
- middlewares

还有一个配置文件

- settings

网易云音乐爬虫代码

items.py

```
# -*- coding: utf-8 -*-

# Define here the models for your scraped items
#
# See documentation in:
# http://doc.scrapy.org/en/latest/topics/items.html

import scrapy


class ProgramItem(scrapy.Item):
    # define the fields for your item here like:
    # name = scrapy.Field()
    program_url=scrapy.Field() #节目链接
    issue=scrapy.Field() #期号
    create_time=scrapy.Field() #创建时间
    play_times=scrapy.Field() #播放次数
    subscription_number=scrapy.Field() #订阅次数
    like_number=scrapy.Field() #点赞数
    comment_number=scrapy.Field() #评论数
    share_number=scrapy.Field() #分享数

class SongItem(scrapy.Item):
    song_id=scrapy.Field() #歌曲编号
    artist_id=scrapy.Field() #歌手编号
    album_id=scrapy.Field() #所属专辑编号
    comment_number=scrapy.Field() #评论数
    title=scrapy.Field() #歌曲名

class LyricItem(scrapy.Item):
    song_id=scrapy.Field() #歌曲编号
    lyric=scrapy.Field() #歌词
```

middlewares.py

```
# -*- coding: utf-8 -*-

# Define here the models for your spider middleware
#
# See documentation in:
# http://doc.scrapy.org/en/latest/topics/spider-middleware.html

from scrapy import signals
from scrapy.http import HtmlResponse
from lxml import etree
import time
from random import choice
from selenium import webdriver
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

ua_list = [
    "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chrom
ium/48.0.2564.82 Chrome/48.0.2564.82 Safari/537.36",
    "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.222
8.0 Safari/537.36",
    "Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.22
14.93 Safari/537.36",
    "Mozilla/5.0 (X11; OpenBSD i386) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/36.0.
1985.125 Safari/537.36",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/32.0.1664.3 Safari/537.36"
]
dcap = dict(DesiredCapabilities.PHANTOMJS)
dcap["phantomjs.page.settings.resourceTimeout"] = 15
dcap["phantomjs.page.settings.loadImages"] = False
dcap["phantomjs.page.settings.userAgent"] = choice(ua_list)


class Music163XihaSpiderMiddleware(object):
    # Not all methods need to be defined. If a method is not defined,
    # scrapy acts as if the spider middleware does not modify the
    # passed objects.

    sleep_seconds = 0.2  # 模拟点击后休眠3秒，给出浏览器取得响应内容的时间
    default_sleep_seconds = 1  # 无动作请求休眠的时间

    def process_request(self, request, spider):
        spider.logger.info('--------Spider request processed: %s' % spider.name)
        page = None
        if 'djradio' in request.url or 'program' in request.url or 'song?id=' in
request.url:
            driver = webdriver.PhantomJS()
            spider.logger.info('--------request.url: %s' % request.url)
            driver.get(request.url)
            driver.implicitly_wait(1)
            # 仅休眠数秒加载页面后返回内容
            time.sleep(self.sleep_seconds)
```

```
            driver.switch_to.frame(driver.find_element_by_name("contentFrame")) # 取得框架内容

            page = driver.page_source
            driver.close()

        elif 'media' in request.url:
            driver = webdriver.PhantomJS()
            spider.logger.info('--------request.url: %s' % request.url)
            driver.get(request.url)
            driver.implicitly_wait(0.2)
            # 仅休眠数秒加载页面后返回内容
            time.sleep(self.sleep_seconds)
            page = driver.page_source
            driver.close()

        return HtmlResponse(request.url, body=page, encoding='utf-8', request=request)
```

spiders.py

```
#-*- coding: utf-8 -*-

import scrapy
import os, json, codecs
from hashlib import md5
from faker import Factory
from music163xiha.items import LyricItem

f = Factory.create()

'''
取得节目列表下所有歌曲信息

提示：歌词 -> 使用 http://music.163.com/api/song/media?id=421203370(歌曲编号)来取得

cd /home/andy/000_music163xiha/scrapy/music163xiha/spiders
pyenv activate scrapy2.7
scrapy crawl songs

'''
class LyricsSpider(scrapy.Spider):
    lyric_url = 'http://music.163.com/api/song/media?id='
    song_list_file='./result/song_url.txt'
    name = "lyrics"
    allowed_domains = ["music.163.com"]

    def __init__(self, *args, **kwargs):
        #从上一步骤取得的 song_list 文件中读取得到所有 url
        f = open(self.song_list_file,"r")
        lines = f.readlines()
        if len(lines)==0:
            self.log('**********\nPlease run spider \'programs\' first\nto get progra
m_url.txt\n**********')
        print '**********'
        print len(lines)
        print '**********'
        song_id_list=[line.split('=')[1] for line in lines]
        song_list=[self.lyric_url+song_id for song_id in song_id_list]
        f.close()
        print '**********'
        print song_list[0]
        print '**********'
        self.start_urls = song_list # 此处应从上一步骤取得的 program_list 文件中读取得到
所有 url



    def parse(self, response):
        self.log('--> url:%s' % response.url)

        result = response.xpath('//body/text()').extract_first()
        json_result = json.loads(result, encoding='utf-8')
        #self.log('--> lyric:%s' % json_result['lyric'])
```

```
        lyric_text=json_result['lyric']#歌词

        lyric=LyricItem()
        lyric['lyric']=lyric_text
        lyric['song_id']=response.url.split('=')[1][:-3]#因为‘=’后面的id后面莫名其妙的跟着3个字母%0A，遂去掉，加[:-3]
        yield lyric
```

虾米音乐爬虫代码

items.py

```
# -*- coding: utf-8 -*-

# Define here the models for your scraped items
#
# See documentation in:
# http://doc.scrapy.org/en/latest/topics/items.html

import scrapy


class SongUrlItem(scrapy.Item):
    # define the fields for your item here like:
    # name = scrapy.Field()
    song_url=scrapy.Field() #歌曲链接

class LyricItem(scrapy.Item):
    # define the fields for your item here like:
    # name = scrapy.Field()
    lyric=scrapy.Field() #歌曲链接
    song_url=scrapy.Field() #歌曲链接


class SongInfoItem(scrapy.Item):
    # define the fields for your item here like:
    # name = scrapy.Field()
    song_url=scrapy.Field() #歌曲链接
    song_title=scrapy.Field() #歌名
    album=scrapy.Field() #专辑
    #singer=scrapy.Field() #歌手
    language=scrapy.Field() #语种
```

spider.py

```
# -*- coding: utf-8 -*-
import scrapy
from xiami.items import LyricItem

class LyricsSpider(scrapy.Spider):
    name='Lyrics'
    allowed_domains=['xiami.com']
    song_url_file='./result/song_url.csv'


    def __init__(self, *args, **kwargs):
        #从song_url.csv 文件中读取得到所有歌曲url
        f = open(self.song_url_file,"r")
        lines = f.readlines()
        #这里line[:-1]的含义是每行末尾都是一个换行符，要去掉
        #这里in lines[1:]的含义是csv第一行是字段名称，要去掉
        song_url_list=[line[:-1] for line in lines[1:]]
        f.close()

        self.start_urls = song_url_list#[:100]#删除[:100]之后爬取全部数据

    def parse(self,response):

        lyric_lines=response.xpath('//*[@id="lrc"]/div[1]/text()').extract()
        lyric=''
        for lyric_line in lyric_lines:
            lyric+=lyric_line
        #print lyric

        lyricItem=LyricItem()
        lyricItem['lyric']=lyric
        lyricItem['song_url']=response.url
        yield lyricItem
```

完整代码稍后会更新到我的github。

遇到的困难和问题解决

被服务器拒绝访问

- 爬虫访问太快，可以在settings.py设置：

  ```
  DOWNLOAD_DELAY = 1
  ```

  还有其它防止被拒绝访问的措施：
- 设置用户代理

  ```
  from faker import Factory
  f = Factory.create()
  USER_AGENT = f.user_agent()
  ```

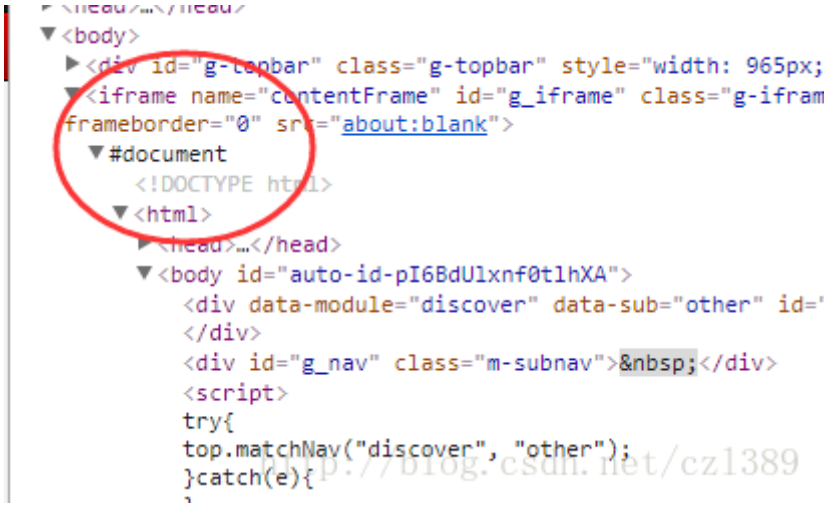
- 设置请求头

```
# Override the default request headers:
DEFAULT_REQUEST_HEADERS = {
    'Host': 'www.xiami.com',
    'Accept': '*/*',
    'Accept-Encoding': 'gzip, deflate, br',
    'Accept-Language': 'zh-CN,zh;q=0.8',
    'Cache-Control': 'no-cache',
    'Connection': 'Keep-Alive',
}
```

xpath爬不到

比如Chrome浏览器会对一些html源码进行规范。有的时候右键“检查”得到的xpath路径是错误的。比如tbody这个标签。

- 网易云音乐有些页面使用了一种叫做iframe 的结构。



抓取其中内容需要，在中间件中做一些工作。

```
def process_request(self, request, spider):
        spider.logger.info('--------Spider request processed: %s' % spider.name)
        page = None
        if 'djradio' in request.url or 'program' in request.url or 'song?id=' in
request.url:
            driver = webdriver.PhantomJS()
            spider.logger.info('--------request.url: %s' % request.url)
            driver.get(request.url)
            driver.implicitly_wait(1)
            # 仅休眠数秒加载页面后返回内容
            time.sleep(self.sleep_seconds)
            driver.switch_to.frame(driver.find_element_by_name("contentFrame")) # 取得框
架内容

            page = driver.page_source
            driver.close()

        elif 'media' in request.url:
            driver = webdriver.PhantomJS()
            spider.logger.info('--------request.url: %s' % request.url)
            driver.get(request.url)
            driver.implicitly_wait(0.2)
            # 仅休眠数秒加载页面后返回内容
            time.sleep(self.sleep_seconds)
            page = driver.page_source
            driver.close()

        return HtmlResponse(request.url, body=page, encoding='utf-8', request=request)
```

In [ ]:

```
```