第 5 关、带参数请求数据

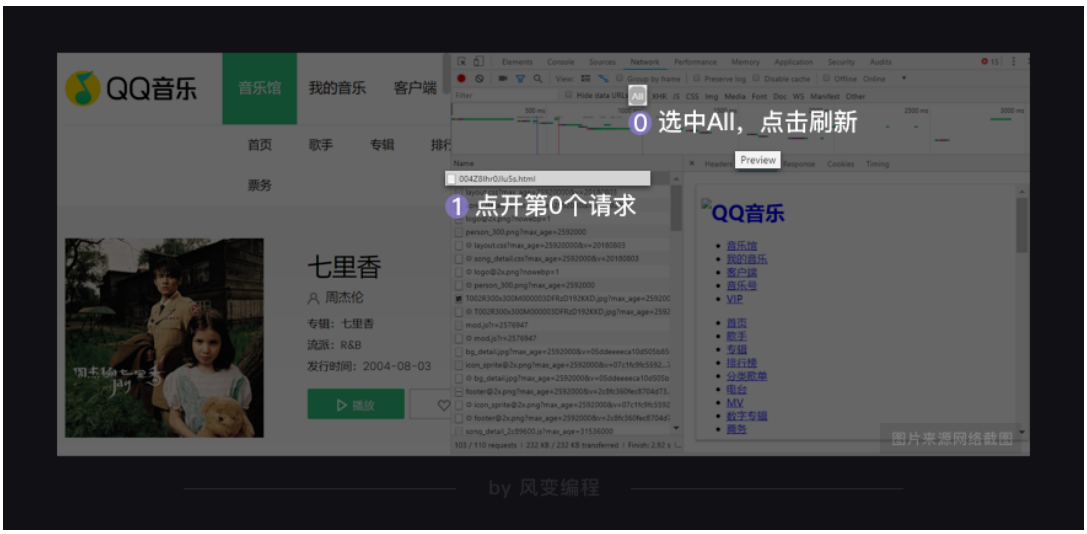
1、带参数请求数据

1-1、什么是带参数请求数据

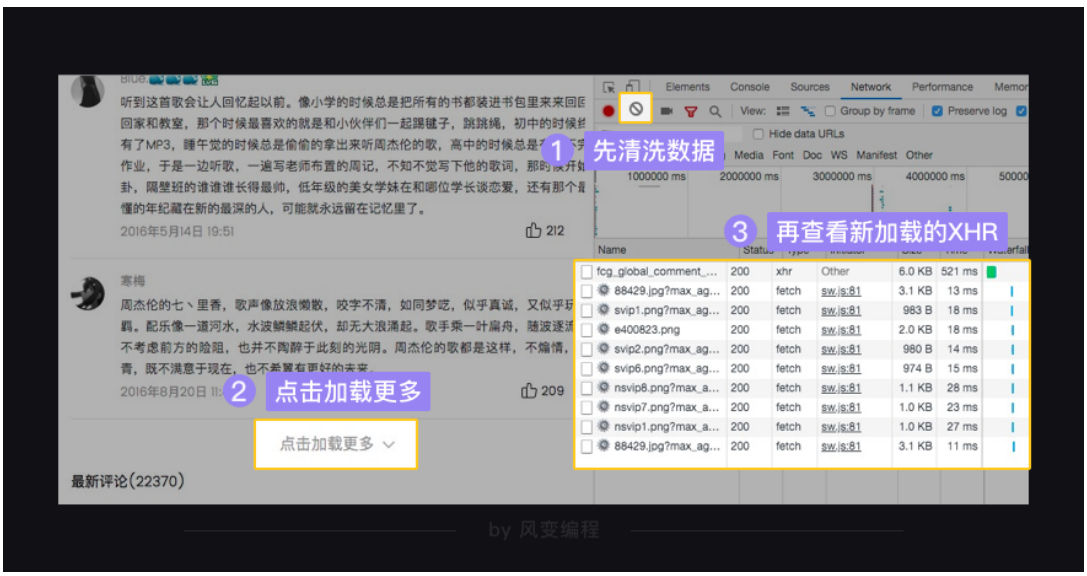
以网址 https://y.qq.com/n/yqq/song/004Z8Ihr0Jlu5s.html 为例，爬取用户的精彩评论：

（1）确定数据所在页面

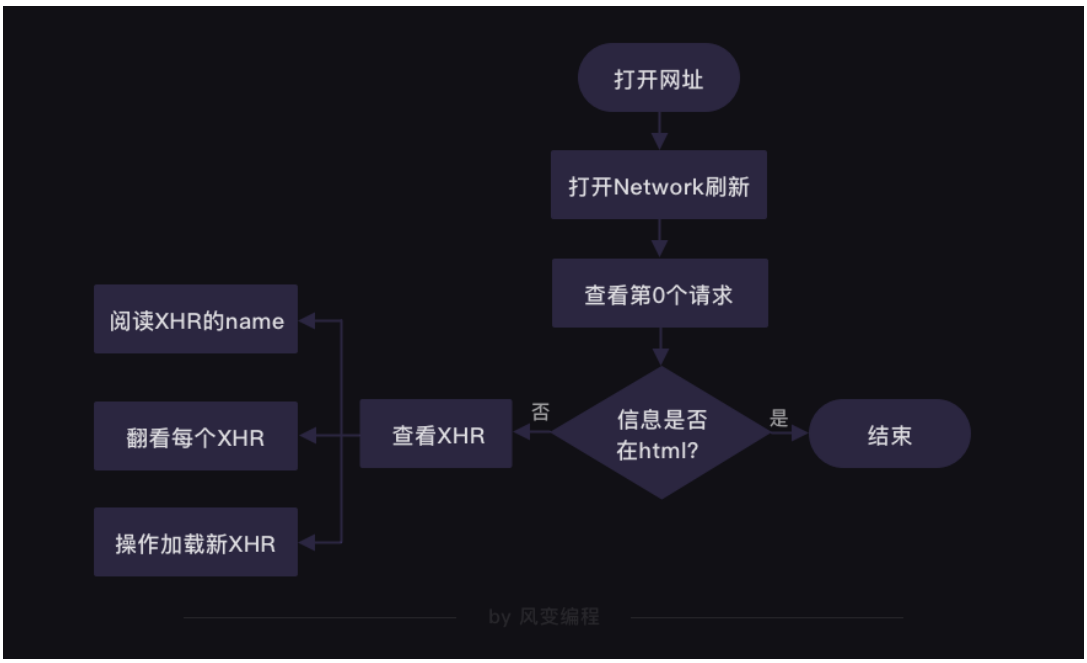
点开第0个请求（第0个请求一般都会是html），没有我们想要的评论信息。



那么就到 XHR 中查找（小 Tips：先把Network面板清空，再点击一下精彩评论的点击加载更多，看看有没有多出来的新XHR，多出来的那一个，就应该是和评论相关）

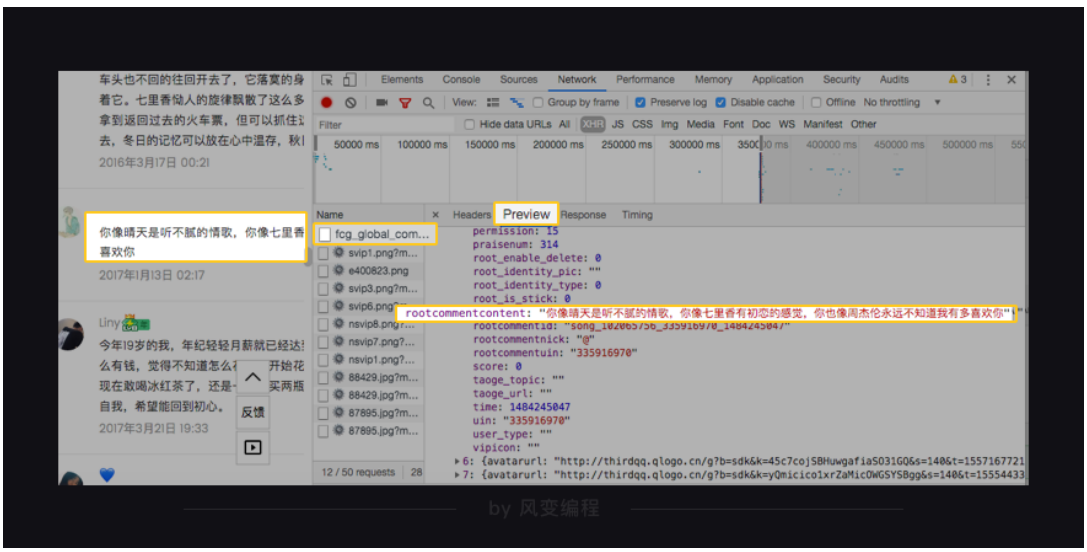


【技巧总结】



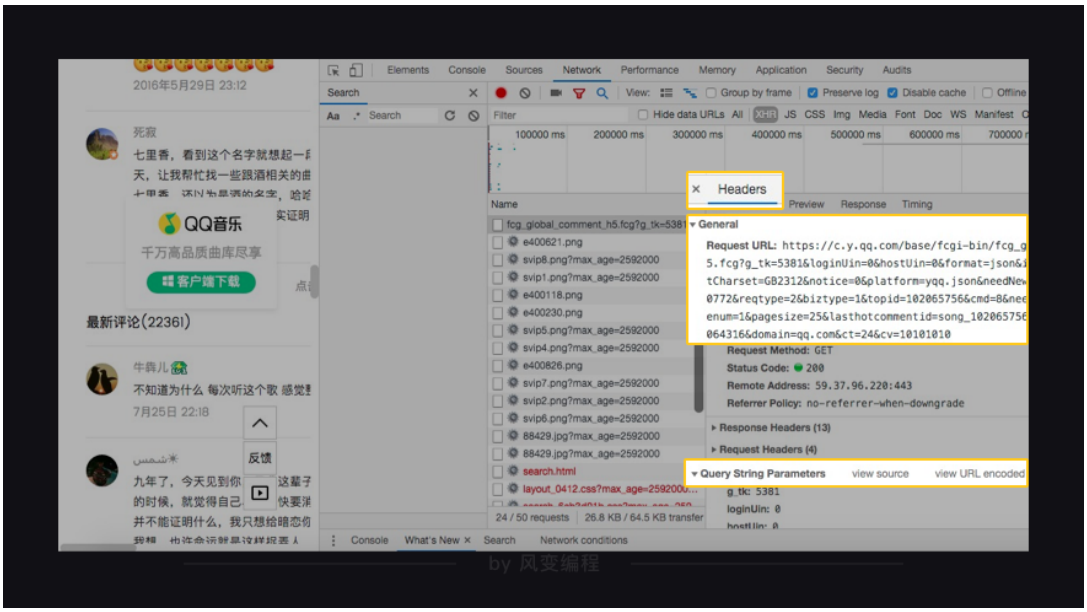
（2）确定数据所在位置

点开这个请求的Preview，能够在 ['comment']['commentlist'] 里找到评论列表。列表的每一个元素都是字典，字典里键 rootcommentcontent 对应的值，就是我们要找的评论。



（3）确定数据所在页面链接

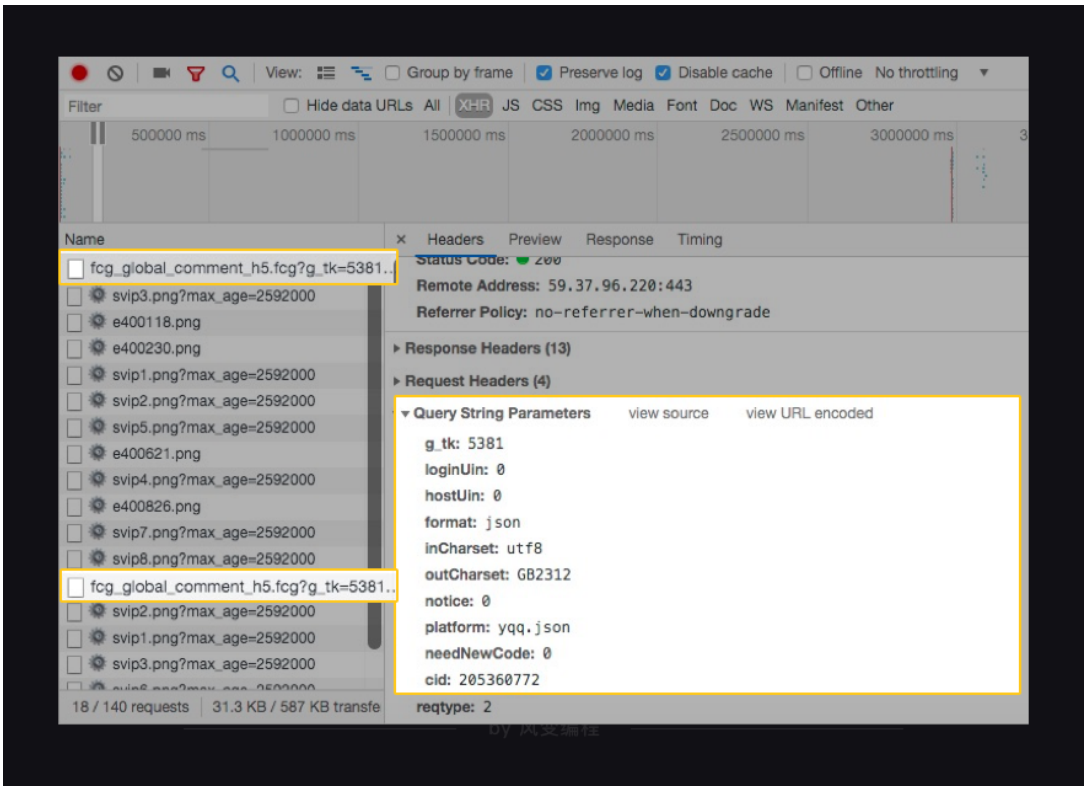
点击请求的Headers栏：General中的Request URL，得到以下链接：https://c.y.qq.com/base/fcgi-bin/fcg_global_comment_h5.fcg?g_tk=5381&loginUin=0&hostUin=0&format=json&inCharset=utf8&outCharset=GB2312&notice=0&platform=yqq.json&needNewCode=0&cid=205360772&reqtype=2&biztype=1&topid=102065756&cmd=6&needmusiccrit=0&pagenum=1&pagesize=15&lasthotcommentid=song_102065756_3202544866_44059185&domain=qq.com&ct=24&cv=10101010



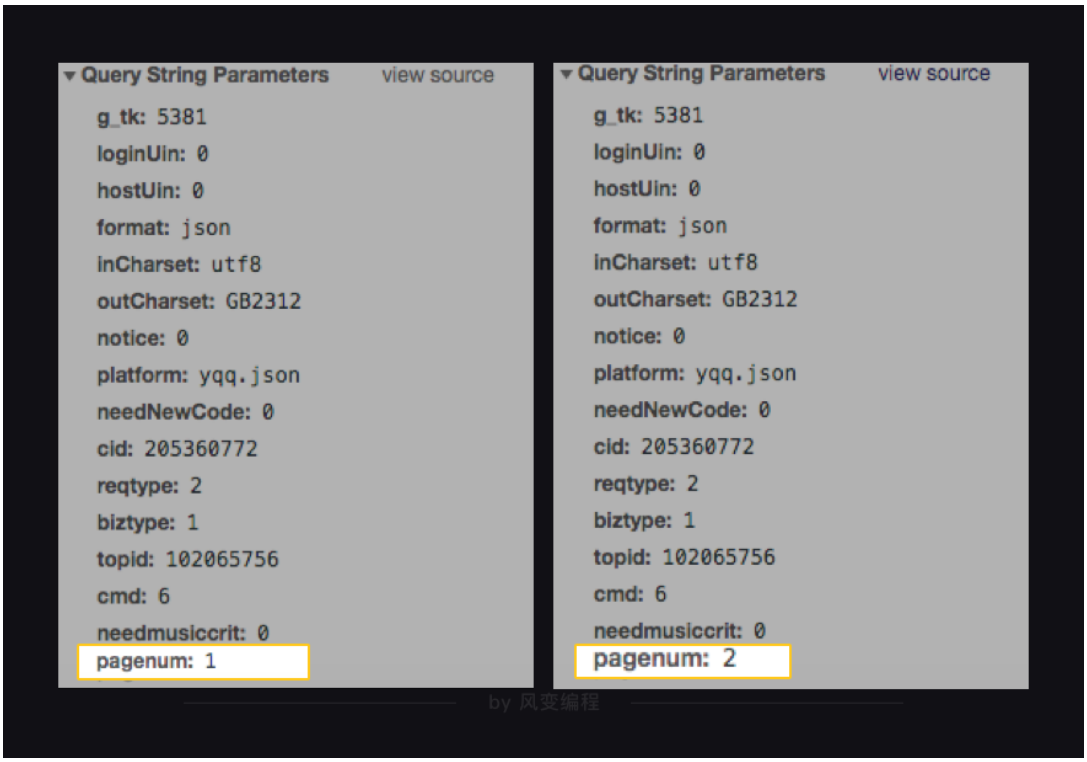
点开 Query String Parametres（查询字符串参数），里面的内容正是链接请求中所附带的参数。

1–2、如何带参数请求数据

点击精彩评论的点击加载更多按钮，此时Network会多加载出更多的XHR，主要关注Name为 fcg_global_comment_h5… 的XHR。



分别点开它们的 Query String Parametres，会发现参数 pagenum 第一次点击加载更多的值为 1，第二第三次点击它的值就变成了 2 和 3 。



也就是说，pagenum=1 等于告诉服务器：我要歌曲信息列表第一页的数据；pagenum=2：我要歌曲信息列表第二页的数据，以此类推 ...

我们只需要写一个循环，每次循环都去更改pagenum的值，这样就能实现爬取好多好多精彩评论。

1-2-1、参数 params

requests模块里的requests.get()提供了一个参数叫params，可以让我们用字典的形式，把参数传进去。

我们可以把Query String Parametres里的内容，直接复制下来，封装为一个字典，传递给params。（注意：要给他们打引号，让它们变字符串）

```
import requests
# 引用requests模块
url = 'https://c.y.qq.com/base/fcgi-bin/fcg_global_comment_h5.fcg'
# 请求歌曲评论的url参数的前面部分

for i in range(5):
    params = {
    'g_tk':'5381',
    'loginUin':'0',
    'hostUin':'0',
    'format':'json',
    'inCharset':'utf8',
    'outCharset':'GB2312',
    'notice':'0',
    'platform':'yqq.json',
    'needNewCode':'0',
    'cid':'205360772',
    'reqtype':'2',
    'biztype':'1',
    'topid':'102065756',
    'cmd':'6',
    'needmusiccrit':'0',
    'pagenum':str(i),
    'pagesize':'15',
    'lasthotcommentid':'song_102065756_3202544866_44059185',
    'domain':'qq.com',
    'ct':'24',
    'cv':'10101010'
    }
    # 将参数封装为字典
    res_comments = requests.get(url,params=params)
    # 调用get方法，下载这个字典
    json_comments = res_comments.json()
    list_comments = json_comments['comment']['commentlist']
    for comment in list_comments:
        print(comment['rootcommentcontent'])
        print('----------------------------------')
```

2、项目：狂热粉丝

2-1、确定目标

（1）目标网站：https://y.qq.com/portal/search.html#page=1&searchid=1&remoteplace=txt.yqq.top&t=song&w=周杰伦；

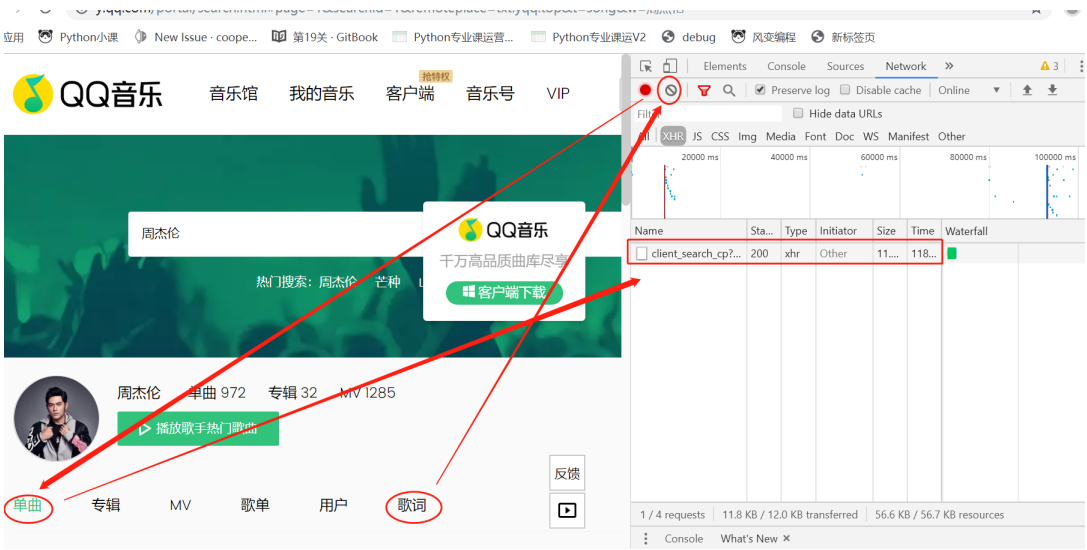
（2）项目目标：爬取周杰伦更多的歌曲信息。

2-2、过程分析

（1）确定数据所在页面

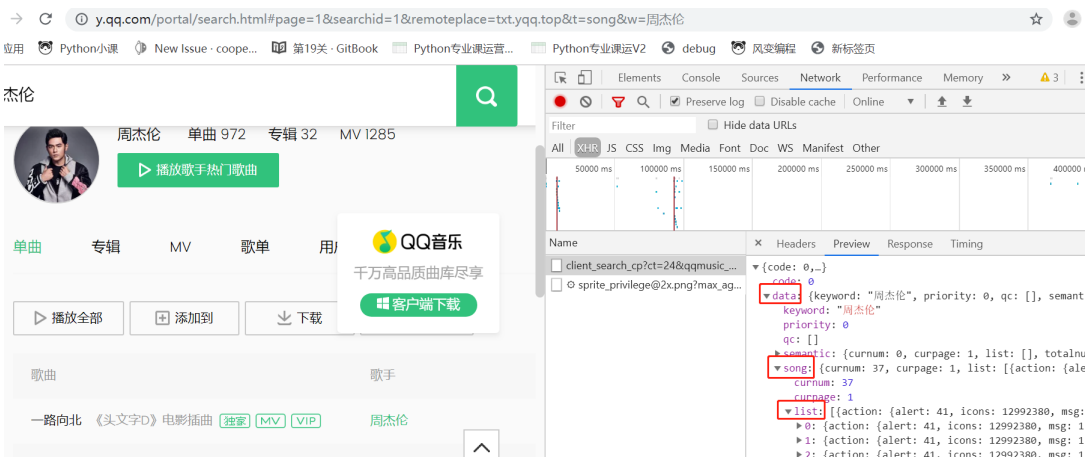
→ 点开第0个请求（第0个请求一般都会是html），没有我们想要的歌曲信息；

→ 那么就到 XHR 中查找（小 Tips：点击【歌曲】，按 F5 刷新页面，再把 NetWork 面板清空，点击【单曲】，出来的就是我们要的单曲信息）



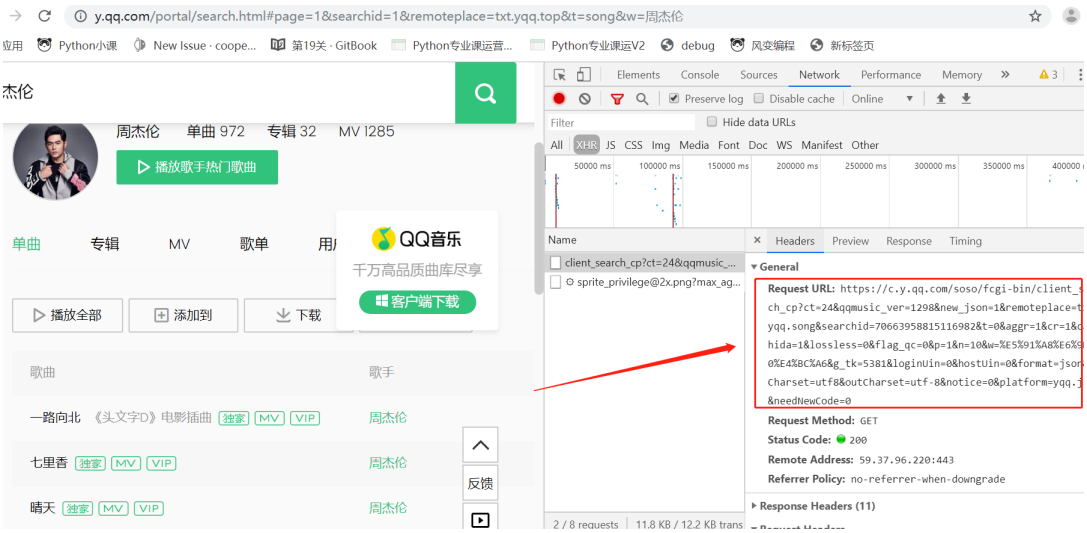
（2）确定数据所在位置

点开这个请求的Preview，能够在 ['data']['song']['list'] 里找到评论列表。列表的每一个元素都是字典，字典里的数据就是我们要找的歌曲信息。



（3）确定数据所在页面链接

点击请求的Headers栏：General中的Request URL，得到以下链接：https://c.y.qq.com/soso/fcgi-bin/client_search_cp?ct=24&qqmusic_ver=1298&new_json=1&remoteplace=txt.yqq.song&searchid=70663958815116982&t=0&aggr=1&cr=1&catZhida=1&lossless=0&flag_qc=0&p=1&n=10&w=周杰伦&g_tk=5381&loginUin=0&hostUin=0&format=json&inCharset=utf8&outCharset=utf-8&notice=0&platform=yqq.json&needNewCode=0



（4）观察页面规律

①、先把 Network 面板清空；②、再修改 page 值按回车键；③、查看 Network 多出来的新 XHR，也就是这个 client_search_cp ...

这个参数是 p，第 1 页XHR的参数p值为 1，第 2、3 页XHR的参数 p 值则为 2 和 3，说明在这个 client_search_cp.. 的请求中，代表页码的参数是 p（page 的缩写）

2-3、代码实现

```
# 直接运行代码就好
import requests
# 引用requests模块
url = 'https://c.y.qq.com/soso/fcgi-bin/client_search_cp'
for x in range(5):
    params = {
    'ct':'24',
    'qqmusic_ver': '1298',
    'new_json':'1',
    'remoteplace':'sizer.yqq.song_next',
    'searchid':'64405487069162918',
    't':'0',
```

```
    'aggr':'1',
    'cr':'1',
    'catZhida':'1',
    'lossless':'0',
    'flag_qc':'0',
    'p':str(x+1),
    'n':'20',
    'w':'周杰伦',
    'g_tk':'5381',
    'loginUin':'0',
    'hostUin':'0',
    'format':'json',
    'inCharset':'utf8',
    'outCharset':'utf-8',
    'notice':'0',
    'platform':'yqq.json',
    'needNewCode':'0'
    }
    # 将参数封装为字典
    res_music = requests.get(url,params=params)
    # 调用get方法，下载这个字典
    json_music = res_music.json()
    # 使用json()方法，将response对象，转为列表/字典
    list_music = json_music['data']['song']['list']
    # 一层一层地取字典，获取歌单列表
    for music in list_music:
    # list_music是一个列表，music是它里面的元素
        print(music['name'])
        # 以name为键，查找歌曲名
        print('所属专辑：'+music['album']['name'])
        # 查找专辑名
        print('播放时长：'+str(music['interval'])+'秒')
        # 查找播放时长
        print('播放链接：
https://y.qq.com/n/yqq/song/'+music['mid']+'.html\n\n')
        # 查找播放链接
```

3、Request Headers

3-1、什么是 Request Headers

每一个请求，都会有一个Requests Headers，我们把它称作请求头。它里面会有一些关于该请求的基本信息，比如：这个请求是从什么设备什么浏览器上发出？这个请求是从哪个页面跳转而来？

- user-agent（中文：用户代理）会记录你电脑的信息和浏览器版本（如我的，就是windows10的64位操作系统，使用谷歌浏览器）；
- origin（中文：源头）和referer（中文：引用来源）则记录了这个请求，最初的起源是来自哪个页面。它们的区别是referer会比origin携带的信息更多些。

如果我们想告知服务器，我们不是爬虫是一个正常的浏览器，就要去修改user-agent。倘若不修改，那么这里的默认值就会是Python，会被浏览器认出来。而对于爬取某些特定信息，也要求你注明请求的来源，即origin或referer的内容。

3-2、如何添加 Request Headers

添加 Request Headers 需要封装一个字典就好了。

而修改origin或referer也和此类似，一并作为字典写入headers就好，拿上面的例子来操作：

```
import requests
url = 'https://c.y.qq.com/soso/fcgi-bin/client_search_cp'

headers = {
    'origin':'https://y.qq.com',
    # 请求来源，本案例中其实是不需要加这个参数的，只是为了演示
    'referer':'https://y.qq.com/n/yqq/song/004Z8Ihr0JIu5s.html',
    # 请求来源，携带的信息比“origin”更丰富，本案例中其实是不需要加这个参数的，只是为了演示
    'user-agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36',
    # 标记了请求从什么设备，什么浏览器上发出
    }
# 伪装请求头

params = {
'ct':'24',
'qqmusic_ver': '1298',
'new_json':'1',
'remoteplace':'sizer.yqq.song_next',
'searchid':'64405487069162918',
't':'0',
'aggr':'1',
'cr':'1',
'catZhida':'1',
```

```
'lossless':'0',
'flag_qc':'0',
'p':1,
'n':'20',
'w':'周杰伦',
'g_tk':'5381',
'loginUin':'0',
'hostUin':'0',
'format':'json',
'inCharset':'utf8',
'outCharset':'utf-8',
'notice':'0',
'platform':'yqq.json',
'needNewCode':'0'
}
# 将参数封装为字典
res_music = requests.get(url,headers=headers,params=params)
# 发起请求，填入请求头和参数
```