

# 第 5 关、带参数请求数据

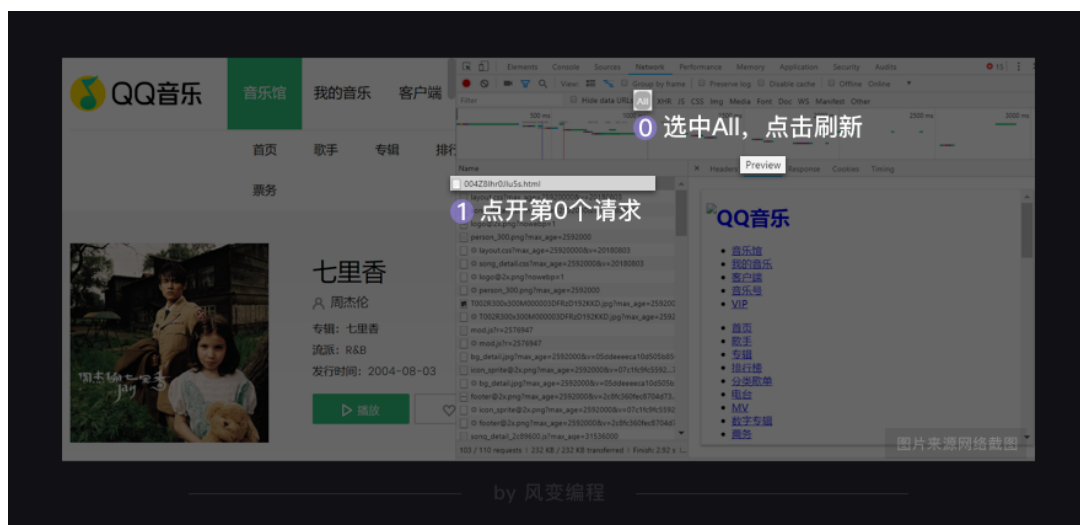
## 1、带参数请求数据

### 1-1、什么是带参数请求数据

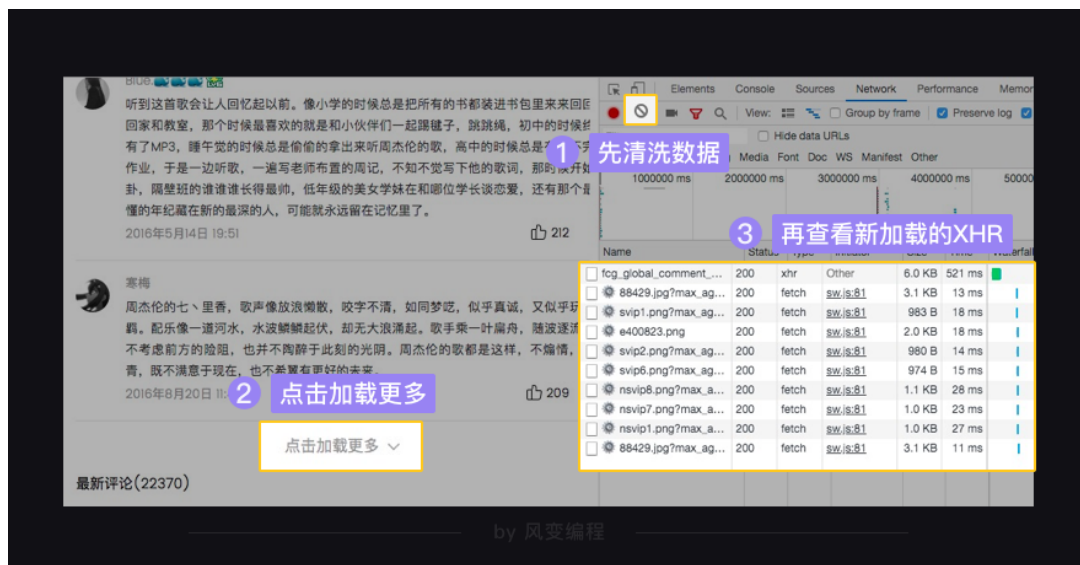
以网址 <https://y.qq.com/n/yqq/song/004Z8lhr0Jlu5s.html> 为例，爬取用户的精彩评论：

(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）

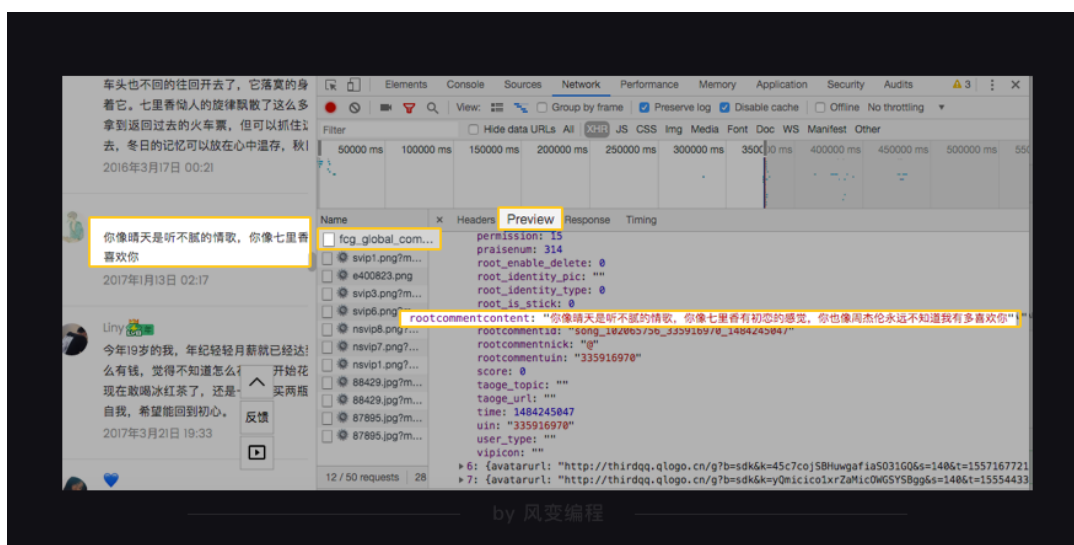


## 【技巧总结】



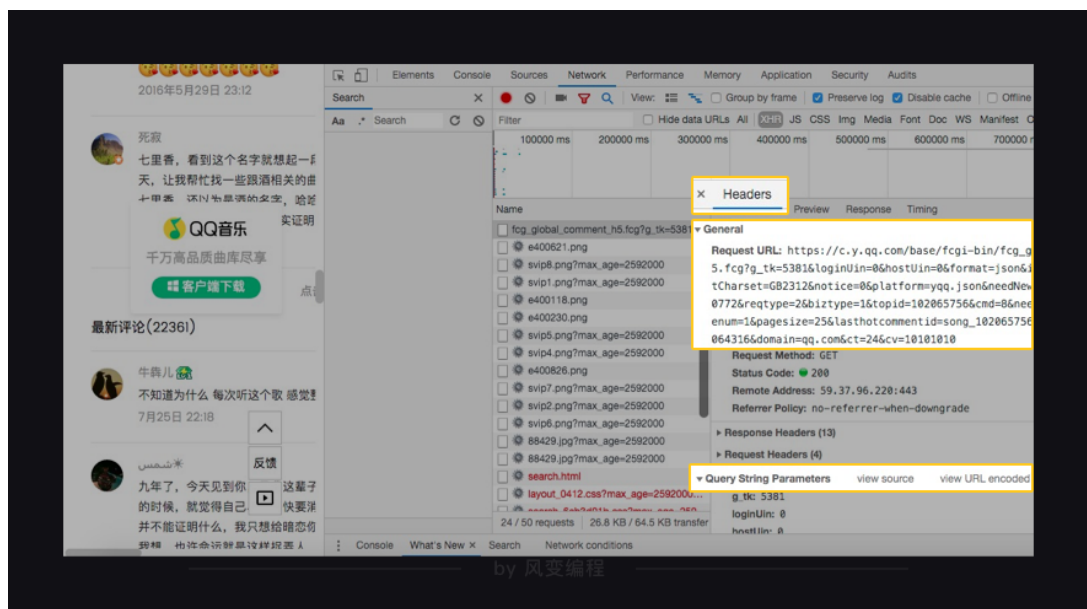
### (2) 确定数据所在位置

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

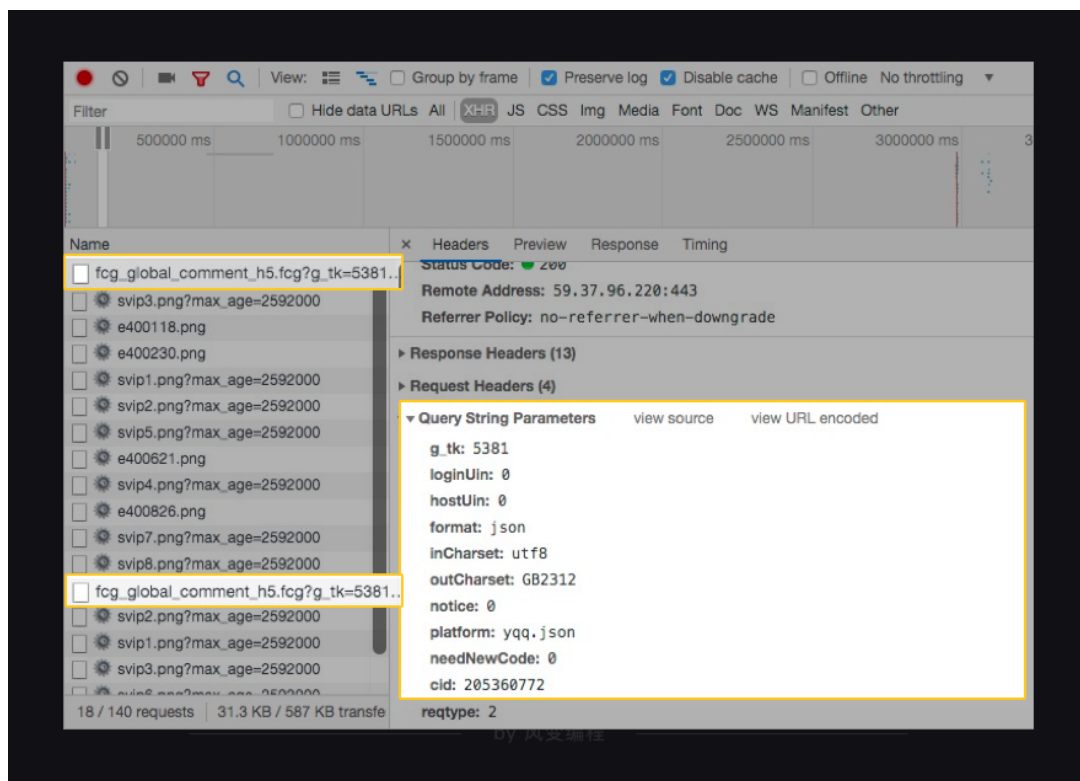
点击请求的Headers栏：General中的Request URL，得到以下链接：[https://c.y.qq.com/base/fcgi-bin/fcg\\_global\\_comment\\_h5.fcgi?g\\_tk=5381&loginUin=0&hostUin=0&format=json&inCharset=utf8&outCharset=GB2312&notice=0&platform=yqq.json&needNewCode=0&cid=205360772&reqtype=2&biztype=1&topid=102065756&cmd=6&needmusiccrit=0&pagenum=1&pagesize=15&lasthotcommentid=song\\_102065756\\_3202544866\\_44059185&domain=qq.com&ct=24&cv=10101010](https://c.y.qq.com/base/fcgi-bin/fcg_global_comment_h5.fcgi?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)



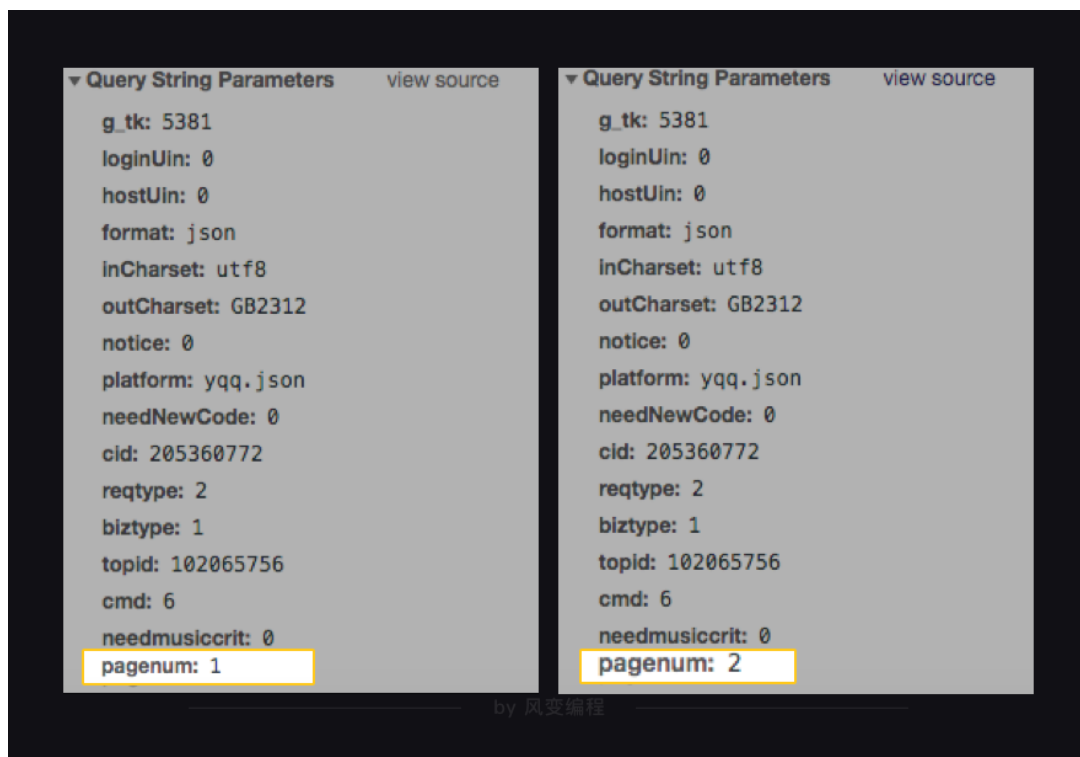
点开 Query String Parametres (查询字符串参数)，里面的内容正是链接请求中所附带的参数。

## 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。

## 传递 URL 参数

你也许经常想为 URL 的查询字符串(query string)传递某种数据。如果你是手工构建 URL，那么数据会以键/值对的形式置于 URL 中，跟在一个问号的后面。例如，`httpbin.org/get?key=val`。Requests 允许你使用 `params` 关键字参数，以一个字符串字典来提供这些参数。举例来说，如果你想传递 `key1=value1` 和 `key2=value2` 到 `httpbin.org/get`，那么你可以使用如下代码：

```
>>> payload = {'key1': 'value1', 'key2': 'value2'}
>>> r = requests.get("http://httpbin.org/get", params=payload)
```

通过打印输出该 URL，你能看到 URL 已被正确编码：

```
>>> print(r.url)
http://httpbin.org/get?key2=value2&key1=value1
```

注意字典里值为 `None` 的键都不会被添加到 URL 的查询字符串里。

你还可以将一个列表作为值传入：

```
>>> payload = {'key1': 'value1', 'key2': ['value2', 'value3']}
>>> r = requests.get('http://httpbin.org/get', params=payload)
>>> print(r.url)
http://httpbin.org/get?key1=value1&key2=value2&key2=value3
```

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）

```
1 import requests
2 # 引用requests模块
3 url = 'https://c.y.qq.com/base/fcgi-bin/fcg_global_comment_h5.fcg'
4 # 请求歌曲评论的url参数的前面部分
5
6 for i in range(5):
7     params = {
8         'g_tk':'5381',
9         'loginUin':'0',
10        'hostUin':'0',
11        'format':'json',
12        'inCharset':'utf8',
13        'outCharset':'GB2312',
14        'notice':'0',
15        'platform':'yqq.json',
16        'needNewCode':'0',
17        'cid':'205360772',
18        'reqtype':'2',
19        'biztype':'1',
20        'topid':'102065756',
21        'cmd':'6',
22        'needmusiccrit':'0',
23        'pagenum':str(i),
24        'pagesize':'15',
25        'lasthotcommentid':'song_102065756_3202544866_44059185',
26        'domain':'qq.com',
27        'ct':'24',
28        'cv':'10101010'
29    }
30    # 将参数封装为字典
31    res_comments = requests.get(url,params=params)
32    # 调用get方法，下载这个字典
33    json_comments = res_comments.json()
34    list_comments = json_comments['comment']['commentlist']
35    for comment in list_comments:
36        print(comment['rootcommentcontent'])
37        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&replace=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

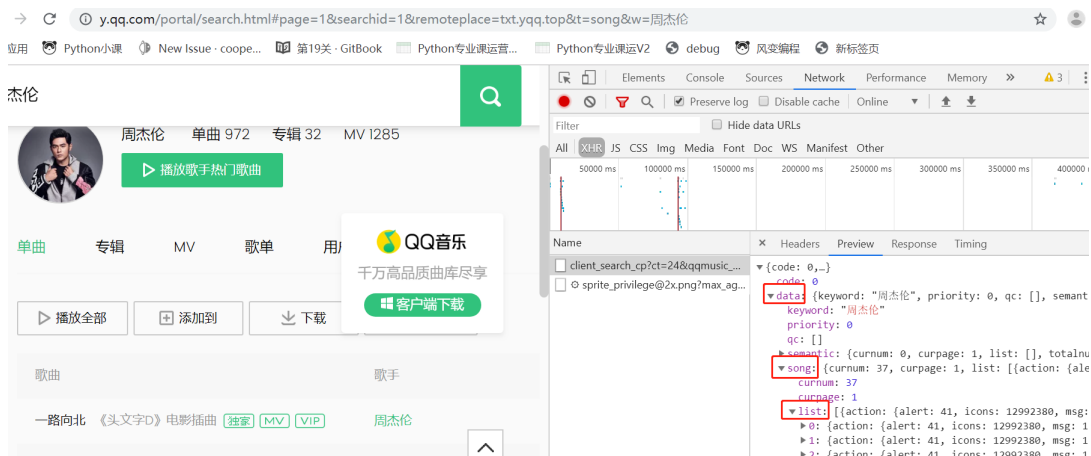
### (1) 确定数据所在页面

- 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】，出来的就是我们想要的单曲信息）



### (2) 确定数据所在位置

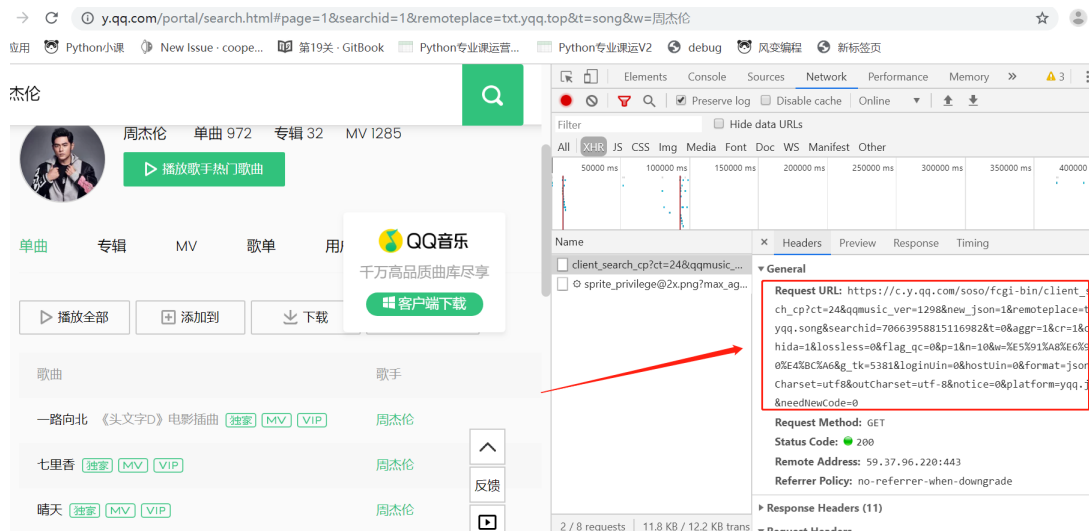
点开这个请求的 Preview，能够在 ['data']['song']['list'] 里找到评论列表。列表的每一个元素都是字典，字典里的数据就是我们要找的歌曲信息。



### (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&noti=0&platform=yqq.json&needNewCode=0](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&noti=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

```

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

```

```

13     'aggr':'1',
14     'cr':'1',
15     'catZhida':'1',
16     'lossless':'0',
17     'flag_qc':'0',
18     'p':str(x+1),
19     'n':'20',
20     'w':'周杰伦',
21     'g_tk':'5381',
22     'loginUin':'0',
23     'hostUin':'0',
24     'format':'json',
25     'inCharset':'utf8',
26     'outCharset':'utf-8',
27     'notice':'0',
28     'platform':'yqq.json',
29     'needNewCode':'0'
30 }
31 # 将参数封装为字典
32 res_music = requests.get(url,params=params)
33 # 调用get方法，下载这个字典
34 json_music = res_music.json()
35 # 使用json()方法，将response对象，转为列表/字典
36 list_music = json_music['data']['song']['list']
37 # 一层一层地取字典，获取歌单列表
38 for music in list_music:
39     # list_music是一个列表，music是它里面的元素
40     print(music['name'])
41     # 以name为键，查找歌曲名
42     print('所属专辑: '+music['album']['name'])
43     # 查找专辑名
44     print('播放时长: '+str(music['interval'])+'秒')
45     # 查找播放时长
46     print('播放链接:
https://y.qq.com/n/yqq/song/'+music['mid']+'.html\n\n')
47     # 查找播放链接

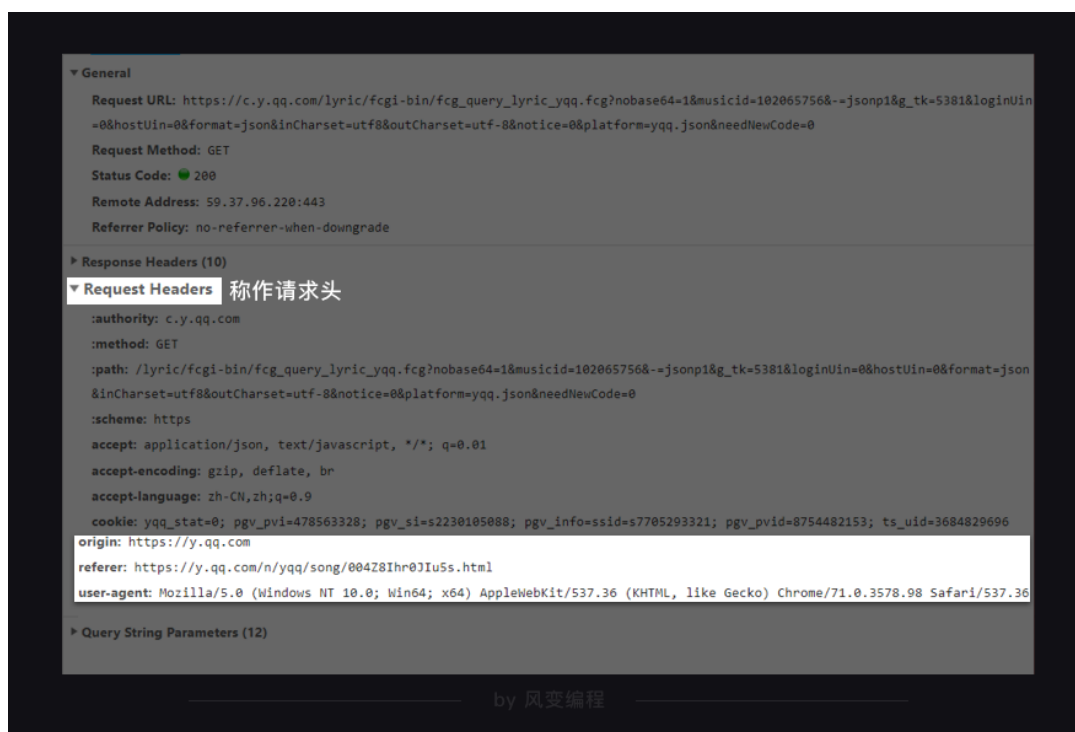
```

## 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。

点击它的官方文档，搜索“user-agent”

## 定制请求头

如果你想为请求添加 HTTP 头部，只要简单地传递一个 `dict` 给 `headers` 参数就可以了。

例如，在前一个示例中我们没有指定 `content-type`:

```
>>> url = 'https://api.github.com/some/endpoint'
>>> headers = {'user-agent': 'my-app/0.0.1'}

>>> r = requests.get(url, headers=headers)
```

注意: 定制 header 的优先级低于某些特定的信息源，例如:

- 如果在 `.netrc` 中设置了用户认证信息，使用 `headers=` 设置的授权就不会生效。而如果设置了 `auth=` 参数，``.netrc`` 的设置就无效了。
- 如果被重定向到别的主机，授权 header 就会被删除。
- 代理授权 header 会被 URL 中提供的代理身份覆盖掉。
- 在我们能判断内容长度的情况下，header 的 `Content-Length` 会被改写。

更进一步讲，Requests 不会基于定制 header 的具体情况改变自己的行为。只不过在最后的请求中，所有的 header 信息都会被传递进去。

注意: 所有的 header 值必须是 `string`、`bytestring` 或者 `unicode`。尽管传递 `unicode` header 也是允许的，但不建议这样做。

by 风变编程

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:

```
1 import requests
2 url = 'https://c.y.qq.com/soso/fcgi-bin/client_search_cp'
3
4 headers = {
5     'origin': 'https://y.qq.com',
6     # 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
7     'referer': 'https://y.qq.com/n/yqq/song/004Z8Ihr0JIu5s.html',
8     # 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
9     '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',
10     # 标记了请求从什么设备，什么浏览器上发出
11 }
12 # 伪装请求头
13
14 params = {
15     'ct': '24',
16     'qqmusic_ver': '1298',
17     'new_json': '1',
18     'remoteplace': 'sizer.yqq.song_next',
19     'searchid': '64405487069162918',
20     't': '0',
21     'aggr': '1',
22     'cr': '1',
23     'catZhida': '1',
```

```
24 'lossless':'0',
25 'flag_qc':'0',
26 'p':1,
27 'n':'20',
28 'w':'周杰伦',
29 'g_tk':'5381',
30 'loginUin':'0',
31 'hostUin':'0',
32 'format':'json',
33 'inCharset':'utf8',
34 'outCharset':'utf-8',
35 'notice':'0',
36 'platform':'yqq.json',
37 'needNewCode':'0'
38 }
39 # 将参数封装为字典
40 res_music = requests.get(url,headers=headers,params=params)
41 # 发起请求，填入请求头和参数
```