

## 第 4 关、json

### 1、Network

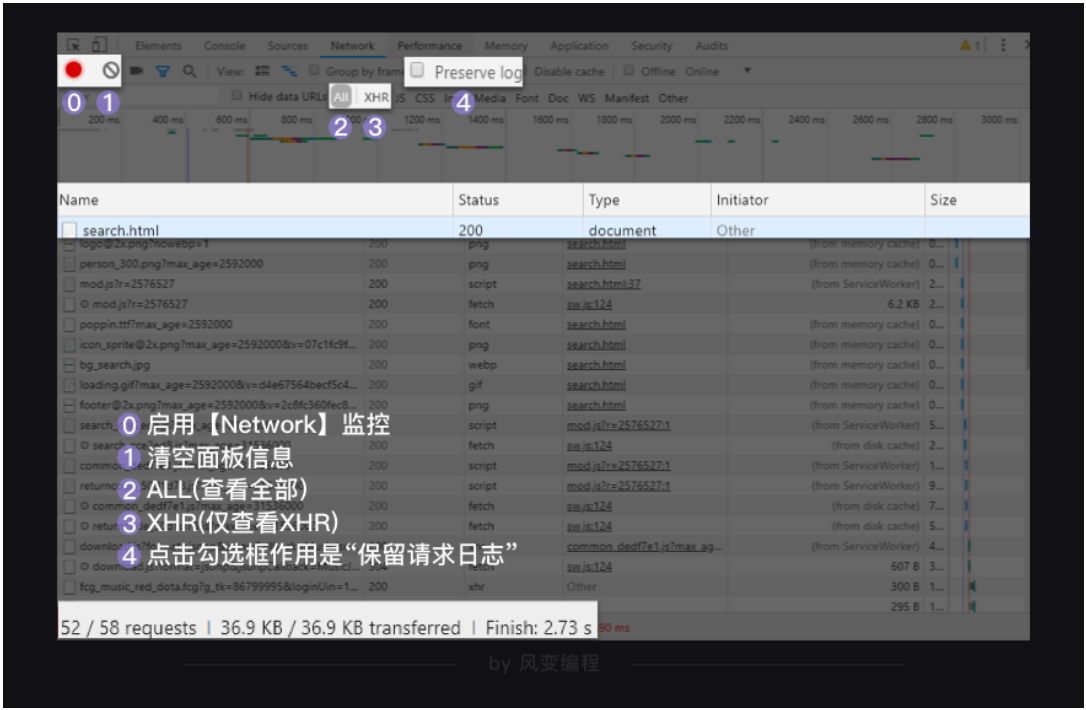
#### 1-1、什么是 Network

Network 的功能是：记录在当前页面上发生的所有请求。Network记录的是实时网络请求。现在网页都已经加载完成，所以不会有东西。（刷新后可弹出）

- 【windows】：在网页的空白处点击右键，然后选择“检查”（快捷方式是ctrl+shift+i），再在 Elements 页面按 ctrl+f；【mac】：在网页的空白处点击右键，然后选择“检查”（快捷键 command + option + I(大写i)）；



#### 1-2、Network 怎么用



- 第1行，是对请求进行分类查看。我们最常用的是：ALL（查看全部）/XHR（仅查看XHR，我们等会重点讲它）/Doc（Document，第0个请求一般在这里），有时候也会看看：Img（仅查看图片）/Media（仅查看媒体文件）/Other（其他）。最后，JS和CSS，则是前端代码，负责发起请求和页面实现；Font是文字的字体；而理解WS和Manifest，需要网络编程的知识。

|             |                     |
|-------------|---------------------|
| ALL         | 查看全部                |
| XHR         | 一种不借助刷新网页即可传输数据的对象  |
| Doc         | Document，第0个请求一般在这里 |
| Img         | 仅查看图片               |
| Media       | 仅查看媒体文件             |
| Other       | 其他                  |
| JS和CSS      | 前端代码，负责发起请求和页面实现    |
| Font        | 字体                  |
| WS和Manifest | 网络编程相关知识，无需了解       |

- 夹在第2行和第1行中间的，是一个时间轴。记录什么时间，有哪些请求。而第2行，就是各个请求，你可以看下面这张表来理解（读，但不需要记忆）。

|           |                       |
|-----------|-----------------------|
| name      | 名字                    |
| status    | 请求的状态，2xx表示成功         |
| type      | 请求的类型（XHR/Doc/Img...） |
| size      | 数据的大小                 |
| time      | 请求的耗时                 |
| waterfall | 瀑布流，用于描述每个请求的起止时间     |

- 第3行，是个统计：有多少个请求，一共多大，花了多长时间。

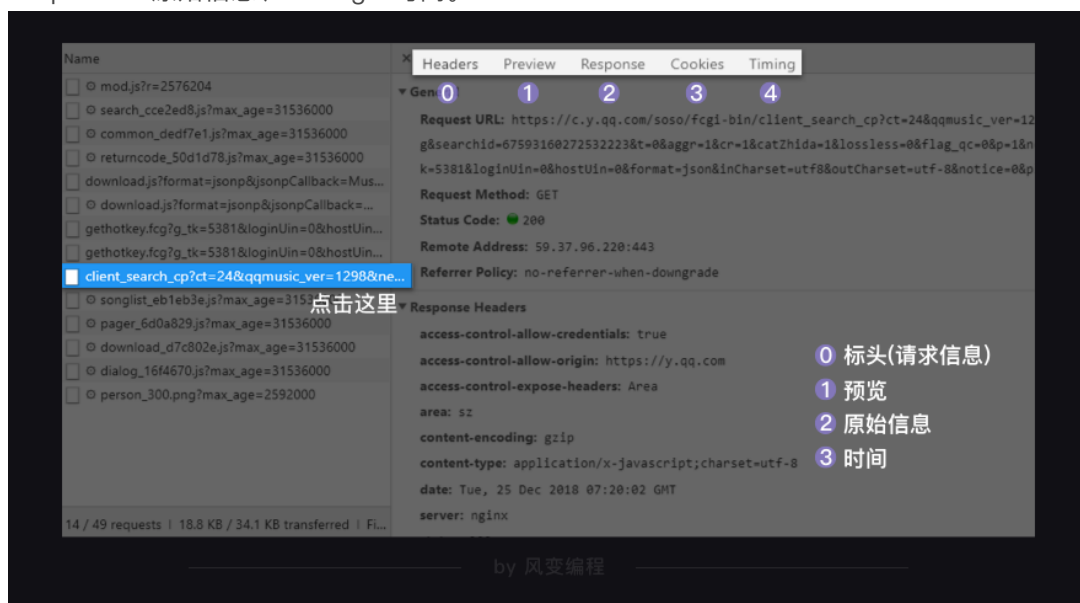
## 2、XHR

### 2-1、什么是 XHR

在Network中，有一类非常重要的请求叫做XHR（当你把鼠标在XHR上悬停，你可以看到它的完整表述是XHR and Fetch）。Ajax技术工作的时候，会创建一个XHR（或是Fetch）对象，实现服务器和浏览器之间传输数据。

### 2-2、XHR 怎么请求

蓝框里面的内容，从左往右分别是：Headers：标头（请求信息）、Preview：预览、Response：原始信息、Timing：时间。



## 3、json

### 3-1、json 是什么？

Python语言当中，json是一种特殊的字符串，这种字符串特殊在它的写法——它是用列表/字典的语法写成的。我们可以将json格式的数据转化为列表/字典，也能将列表/字典转为json格式的数据。

```
1 a = '1,2,3,4'
2 # 这是字符串
3 b = [1,2,3,4]
4 # 这是列表
5 c = '[1,2,3,4]'
6 # 这是字符串，但它是用json格式写的字符串
```

### 3-2、json 数据如何解析？



```

9 # 查找class属性值为“js_song”的a标签，得到一个由标签组成的列表
10 print(list_music)
11 # 打印它

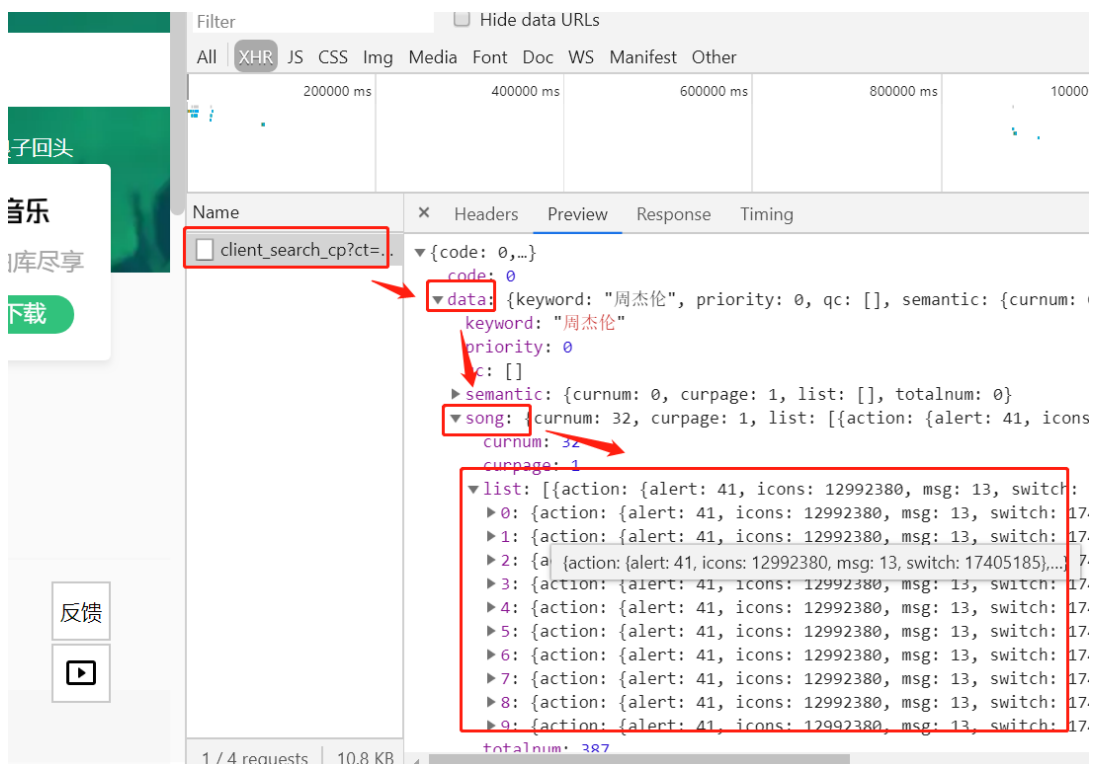
```

结果为空，打印源代码后发现歌曲名不存在与 html 页面。

## (2) XHR 中查找请求文件

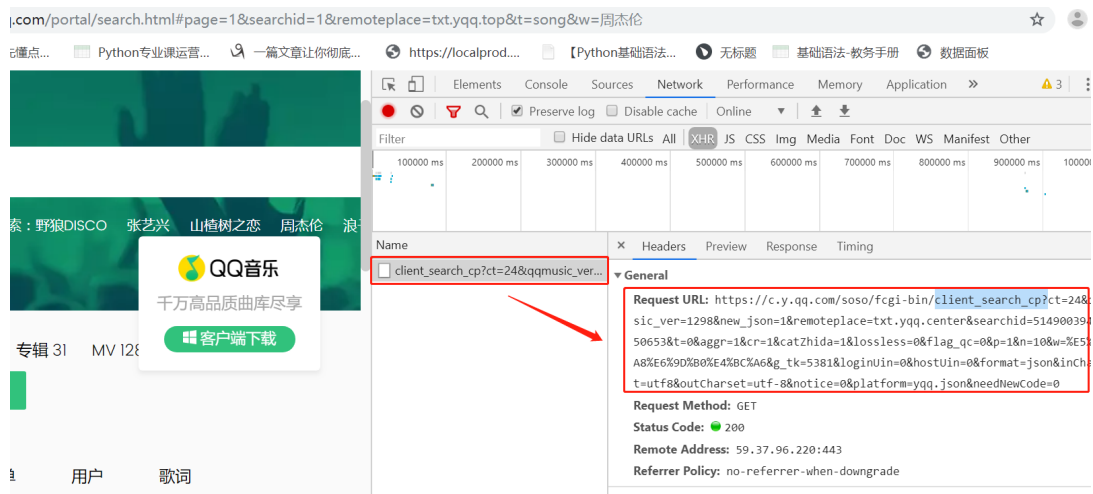


## (3) 数据在 client\_search\_cp?... 请求文件 ['data']['song']['list'] 中



## 4-3、代码实现

### 4-3-1、获取请求地址



```
1 import requests
2 # 引用requests库
3 res_music = requests.get('https://c.y.qq.com/soso/fcgi-
  bin/client_search_cp?
  ct=24&qqmusic_ver=1298&new_json=1&remoteplace=txt.yqq.song&searchid=609974
  26243444153&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')
4 # 调用get方法，下载这个字典
```

#### 4-3-2、解析数据

json() 将获取到的 Response 对象转换为字典，再根据歌曲名的位置，我们用字典取值的方式找到页面上的歌曲名信息。

```
1 import requests
2 # 引用requests库
3 res_music = requests.get('https://c.y.qq.com/soso/fcgi-
  bin/client_search_cp?
  ct=24&qqmusic_ver=1298&new_json=1&remoteplace=txt.yqq.song&searchid=609974
  26243444153&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')
4 # 调用get方法，下载这个字典
5 json_music = res_music.json()
6 # 使用json()方法，将response对象，转为列表/字典
7 list_music = json_music['data']['song']['list']
8 # 一层一层地取字典，获取歌单列表
9 for music in list_music:
10 # list_music是一个列表，music是它里面的元素
11     print(music['name'])
12     # 以name为键，查找歌曲名
```