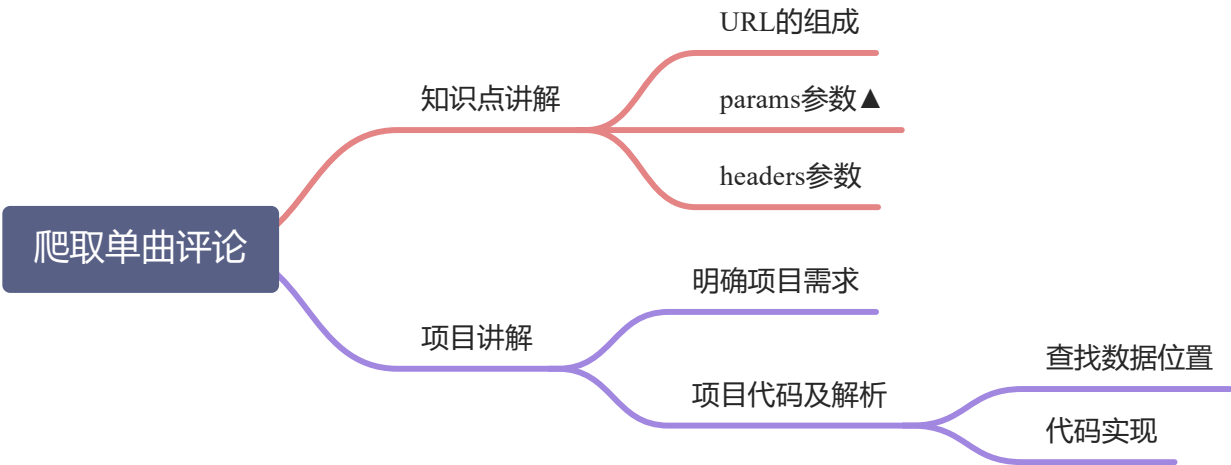


# 第5课 带参数请求

## 一、课程结构导图



注：▲为重点知识点。

## 二、知识点讲解

### 2.1 URL组成

**概念：**URL中文译为统一资源定位符，是 Internet 上资源的地址，比如一个文本文件，一张图片，一个视频。通过 URL 我们可以知道网络资源的位置以及访问它的协议。

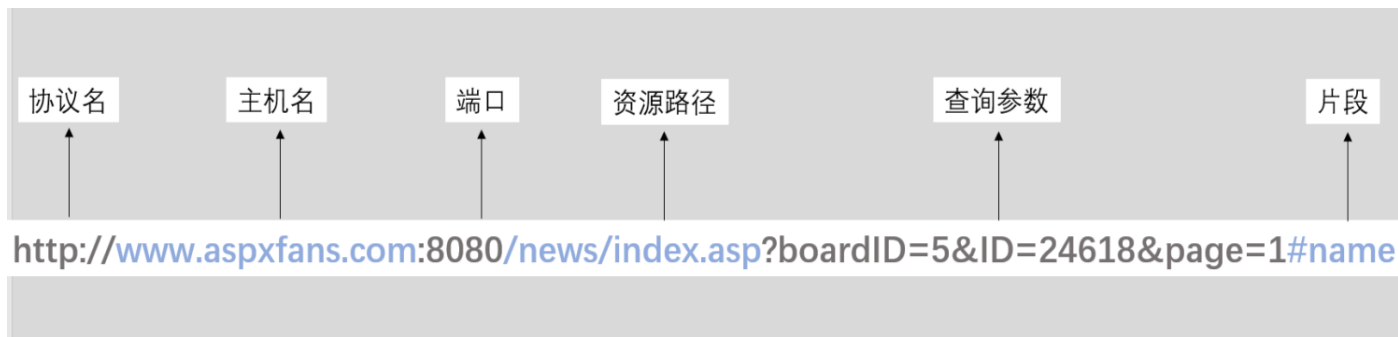
**组成：** http(s)://<host>:<port>/<path>?<query>#<frag>

英文名	含义
http或https	协议名称
<host>	主机名，可以是域名或主机ip地址
<port>	端口，默认端口是80或433（可选）
<path>	资源路径（可选）
<query>	查询参数，多个参数使用&分隔（可选）

<frag>

片段，例如某一篇文章内容（可选）

示例：



## 2.2 params参数

**使用场景：**params是requests模块里的requests.get()的一个参数。一般将url中的查询参数（query）以字典形式封装，并传递给params参数，进行请求操作。

**用法：**requests.get(url,params=params)，一般用于自定义url。

示例：

```
1 import requests
2 payload = {'key1': 'value1', 'key2': 'value2'}
3 r = requests.get("http://httpbin.org/get", params=payload)
4 print(r.url)
5
6 # 打印结果为
7 # http://httpbin.org/get?key2=value2&key1=value1
```

参数内容查找：

1. 点击请求

2. 点击Headers

3. 点击Query String Parameters

params内容

### 注意点:

1. 当我们需要使用params参数请求数据时, 可以把Query String Parameters里的内容, 直接复制下来, 封装为一个字典, 传递给params。要特别注意: 要给他们打引号, 让它们变字符串。
2. 在使用params参数时, 需要给params参数赋值。例如data = {'key1': 'value1', 'key2': 'value2'}, 请求语句应写为: requests.get(url, params=data)。
3. 我们可以通过观察每个请求中的Query String Parameters的内容变化, 寻找出url的变化规律, 一般用于翻页爬取。

## 2.3 headers参数

**使用场景:** 服务器会通过请求头判断访问者是一个普通的用户(通过浏览器), 还是一个爬虫者(通过代码)。我们可以通过headers参数来增加请求头中的headers信息, 伪装成普通的用户, 避免出现被反爬的情况。

**用法:** requests.get(url, headers=headers)。

**示例:**

```

1 import requests
2 url = 'https://c.y.qq.com/soso/fcgi-bin/client_search_cp'
3 headers = {
4     'origin': 'https://y.qq.com',
5     'referer': 'https://y.qq.com/n/yqq/song/004Z8Ihr0JIu5s.html',
6     'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
7 }
8 r = requests.get(url, headers=headers)
9 print('伪装成功!')

```

## 参数内容查找：

1. 点击请求

2. 点击Headers

3. 点击Request Headers

headers参数内容

origin: https://y.qq.com

referer: https://y.qq.com/n/yqq/song/004Z8Ihr0JIu5s.html

user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36

## 注意点：

1. 一般情况下，在设置headers参数时，我们只需要设置user-agent（用户代理）即可。
2. origin（源头）和referer（引用来源）的区别是referer会比origin携带的信息更多些。

# 三、项目讲解

## 3.1 明确项目需求

**需求：**打开QQ音乐网站，爬取周杰伦《七里香》单曲中的歌曲精彩评论。

**歌曲评论图示：**

精彩评论(310)



知非 置顶

这首歌唤醒了所有关于你的记忆，我又看见了你帮我画手抄报的侧颜，听见了我们九月里散步踩碎落叶的响声，闻到了你最爱的草莓冰激凌的味道，摸到了我织给你的那条软软的围巾。只可惜啊手抄报被老师收走了，落叶终究归根，听说你后来也不吃草莓味的冰激凌了，那条围巾你估计也埋在了看不到的地方。关于你，我好像什么都没有了。

3月13日 23:23

4973



青莲很忙 置顶

那年夏天的黄昏将泥砌的墙涂成炽热颜色，窗外麻雀窃窃私语着什么，院内梧桐叶层层叠叠，我忧心的麻雀忘记物理所学导线两端无电压，我念着你忘了一切。

2018年11月10日 01:33

6291



蜗牛 置顶

七月的知了将夏天唤醒，手中的铅笔在稿纸上肆意的涂鸦。房间里散发着栀子花香，微风轻轻拂动淡色的窗帘。听着鸟鸣与微风，指尖拨动着音弦，流年在指缝中轻捻。夕阳细碎的暖光，风轻云淡，夕光伴影。时空中夹杂着七里香的流年，而我双眸却全是你的画面。

2018年11月10日 10:46

举报 4291

## 3.2 项目代码及解析

### 3.2.1 查找数据位置

**数据查找技巧图示：**

**目的1：**判断评论数据是否在html。

**关键操作1：**打开检查-点击All-点击第0个请求-点击Perview。

**操作图示1：**

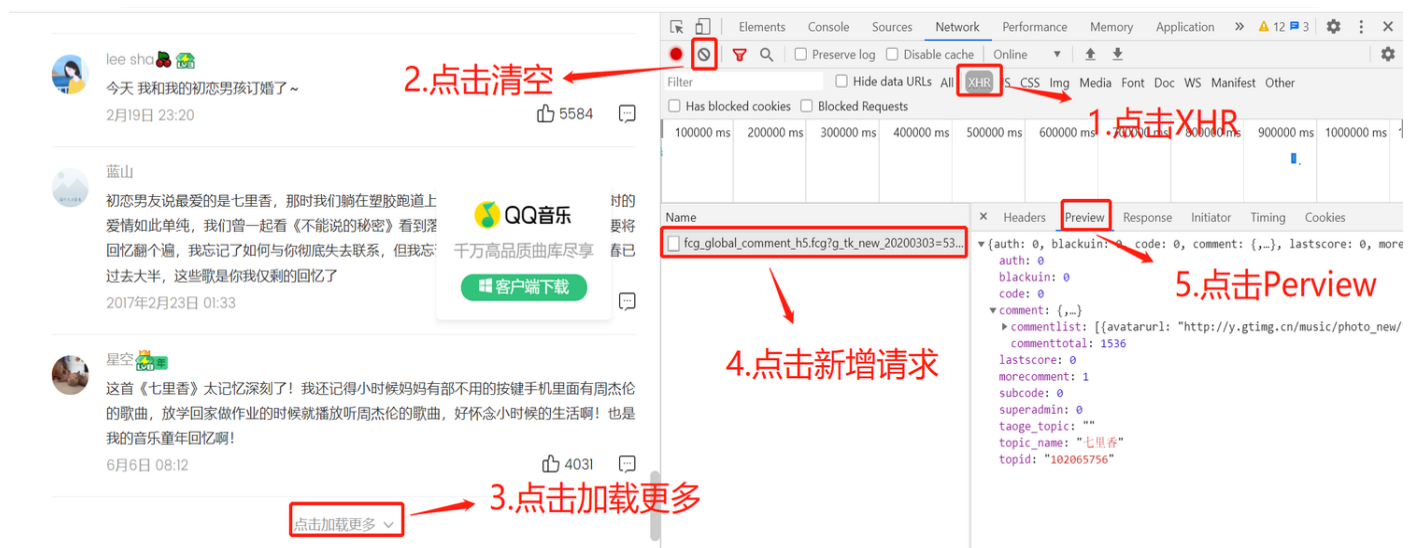
## 精彩评论 (1536)



目的2：判断评论数据是否在XHR请求。

关键操作2：点击XHR-清空所有请求-点击加载更多-点击新增请求-点击Perview。

操作图示2：



注意点：

1. 当我们寻找一个XHR请求时，可先通过清空所有请求，再进行点击操作，观察新增的请求，来查找出数据。



2. 通过对比，证明了评论的数据不在html中，而是在XHR请求中。

### 3.2.2 代码实现

**目的：**使用params参数和headers参数请求，爬取歌曲前五页的精彩评论。

**关键代码：**

```
1 import requests
2 url = 'https://c.y.qq.com/base/fcgi-bin/fcg_global_comment_h5.fcg'
3 # 请求歌曲评论的url参数的前面部分
4
5 # 伪装普通用户
6 headers = {
7     'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
8     }
9
10 # 将查询参数封装为字典
11 for i in range(5): # 翻页5次
12     params = {
13         'g_tk': '5381',
14         'loginUin': '0',
15         'hostUin': '0',
16         'format': 'json',
17         'inCharset': 'utf8',
18         'outCharset': 'GB2312',
19         'notice': '0',
20         'platform': 'yqq.json',
21         'needNewCode': '0',
22         'cid': '205360772',
23         'reqtype': '2',
24         'biztype': '1',
25         'topid': '102065756',
26         'cmd': '6',
27         'needmusiccrit': '0',
28         'pagenum': str(i),
29         'pagesize': '15',
30         'lasthotcommentid': 'song_102065756_3202544866_44059185',
31         'domain': 'qq.com',
32         'ct': '24',
33         'cv': '10101010'
34     }
35
```

```
36     # 使用params参数和headers参数请求数据
37     res_comments = requests.get(url,params=params,headers=headers )
38     json_comments = res_comments.json()
39     list_comments = json_comments['comment']['commentlist']
40     for comment in list_comments:
41         print(comment['rootcommentcontent']) #打印出歌曲的精彩评论
```

#### 代码解析：

1. 第28行代码，代码行'pagenum':str(i)，需要对i变量转为字符串，而不能直接写为'pagenum':i。
2. 第39行代码，展开数据层级观察，我们会发现歌曲的评论语在提取['comment']['comlist']后的列表里，所以通过json()方法将json数据转为列表/字典，逐层提取出歌曲的评论语所在的列表(list\_comment)。
3. 第40行代码，最后只需通过for循环遍历列表内容，提取歌曲评论与即可。