**《课程名称》 课 程 论 文**

**(2018-2019 学年第 二 学期)**

**美团美食页爬虫设计**

**学生姓名：郑宜静 陈丹燕 陈泓宇 杨官梅**

**提交日期： 年 月 日 学生签名：**

|  |  |  |  |
| --- | --- | --- | --- |
| **学 号** |  | **座位编号** |  |
| **学 院** |  | **专业班级** |  |
| **课程名称** |  | **任课教师** |  |
| **教师评语：** | | | |
| **本论文成绩评定：** **分** | | | |

美团美食页爬虫设计

**郑宜静、****陈丹燕、陈泓宇、杨官梅**

**摘要：**本文档为针对美团美食爬虫项目的总结报告，爬虫程序采用python构建，爬取信息包括地区的美食列表、美食关键字搜索结果列表、城市信息及城市地段美食分布、店铺基本信息、店铺推荐菜、店铺评论等，实现对美团美食相关信息的完整收集，并对美团美食网页的反爬虫系统和可行的对应的反反爬虫策略进行分析和总结。

**关键词：**美团；爬虫；反爬虫；反反爬虫；验证码；XHR；加密；

1. **引言**

在大数据时代下，信息和数据变得越来越有价值，于是越来越多的网络爬虫出现，这种行为一方面窃取了许多的公司和网站的信息，另一方面提高了数据服务器的负载压力。针对这一情况，许多公司都进行了反爬虫检测。本文针对美团的美食相关部分的多个页面，对美团网站的反爬虫策略进行了探索以及提出相应的反反爬虫策略。

本文档首先对当前存在的爬虫反爬虫策略进行了解，然后详细分析了美团美食相关的页面组成、数据结构、请求头部等，接着根据自身以及网上的知识总结对应的方法，再根据现有的条件选取反反爬虫策略进行测试，最后展示测试结果。

1. **预备工作**
   1. **网页爬虫现状调查**

****

* 1. **确立爬虫目标**

对美团的美食等相关页面进行分析和爬虫，获得不同区域美食商铺列表、美食评论列表、美食搜索结果列表、城市相关信息、美食地域分布情况等信息，并提取和记录在csv文件中。

* 1. **美团美食的网页分析**
     1. **PC端美食主页**
        1. **请求分析**
           1. **Request URL**

http://www.meituan.com/meishi/api/poi/getPoiList?cityName=%E5%B9%BF%E5%B7%9E&cateId=0&areaId=0&sort=&dinnerCountAttrId=&page=1&userId=&uuid=4613f329-a16a-48c2-8cdf-d91df2e6bdef&platform=1&partner=126&originUrl=http%3A%2F%2Fwww.meituan.com%2Fmeishi%2F&riskLevel=1&optimusCode=10&\_token=eJxVjstygjAUht8l2zKSEBLEna2iscYbKGrHBQoCAppyEaTTd286YxedOTP%2F5XyL%2FwvkzAc9BKEJoQLuQQ56AHVghwIFlIX8EIqwDolGuhgp4PS%2FMw1DAcd8MwC9D2J2FQOjw2%2BxkvkDEUyVLtUPytNiaTVd3i%2FDJAKishQ9Va3rupMFcVl5187plqnSF1GsyglAopkjUanJU72nln%2BZy82SLeLwKl0wqdPLGr7Xl%2F4yCtSY5aW3pu4tYa4%2BLGz%2FtYmP8VLX7H3Kp5Yx8vQwd164OZgKOtZVzGtrpFlJu24gU2%2Fj5By%2BLch9Rs9EX%2BztZEhGmC%2BdXbwdsM9Bwl6EpkFvseIrb9dG9gKTmD3yeUnHPBArwvjOF9a28ecWzTf9uW%2B%2F84k9uaHKvWwhe6OPabt3qspooiuOLCvjjzBVz23mzHal25pa1924oUtD90GpSAVC6QYJ2ojCaY4%2B6VezYpjuwfcP%2FVeIFA%3D%3D

* + - * 1. **Request Method:** GET
        2. **Request Headers**

**Accept:** application/json

**Accept-Encoding:** gzip, deflate

**Accept-Language:** zh,zh-TW;q=0.9,en-US;q=0.8,en;q=0.7,zh-CN;q=0.6

**Connection:** keep-alive

**Cookie:** \_lxsdk\_cuid=16b34d7ff07c8-07b64189060e93-e353165-144000-16b34d7ff08c8; ci=20; \_hc.v=6bd5e373-9e29-674d-16ed-e3a84a9dce7b.1559968514; iuuid=AD87C2B2B6FB5F8D19B198B95CA4FEBBC4DF9E57BEF92F3FA2087B1803E1F51E; cityname=%E5%B9%BF%E5%B7%9E; \_lxsdk=AD87C2B2B6FB5F8D19B198B95CA4FEBBC4DF9E57BEF92F3FA2087B1803E1F51E; webp=1; \_\_utmz=74597006.1560040810.1.1.utmcsr=blog.csdn.net|utmccn=(referral)|utmcmd=referral|utmcct=/xing851483876/article/details/81842329; \_\_utma=74597006.648188052.1560040810.1560228630.1560229934.3; i\_extend=C\_b0E280564580298145705312641275118984368883\_e849009944306480013\_a%e6%97%a9%e8%8c%b6Gimthomepagecategory11H\_\_a100173\_\_b3; client-id=c86681bd-284d-4657-8ce3-7c694f8f3422; uuid=4613f329-a16a-48c2-8cdf-d91df2e6bdef; \_lx\_utm=utm\_source%3Dblog.csdn.net%26utm\_medium%3Dreferral%26utm\_content%3D%252Fxing851483876%252Farticle%252Fdetails%252F81842329; lat=22.987182; lng=113.345815; \_lxsdk\_s=16b87252e1a-961-468-9f0%7C%7C4

**Host:** www.meituan.com

**Referer:** http://www.meituan.com/meishi/

**User-Agent:** Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36

* + - * 1. **Query String Parameters**

**cityName:** 广州

**cateId:** 0

**areaId:** 0

**sort:**

**dinnerCountAttrId:**

**page:** 1

**userId:**

**uuid:** 4613f329-a16a-48c2-8cdf-d91df2e6bdef

**platform:** 1

**partner:** 126

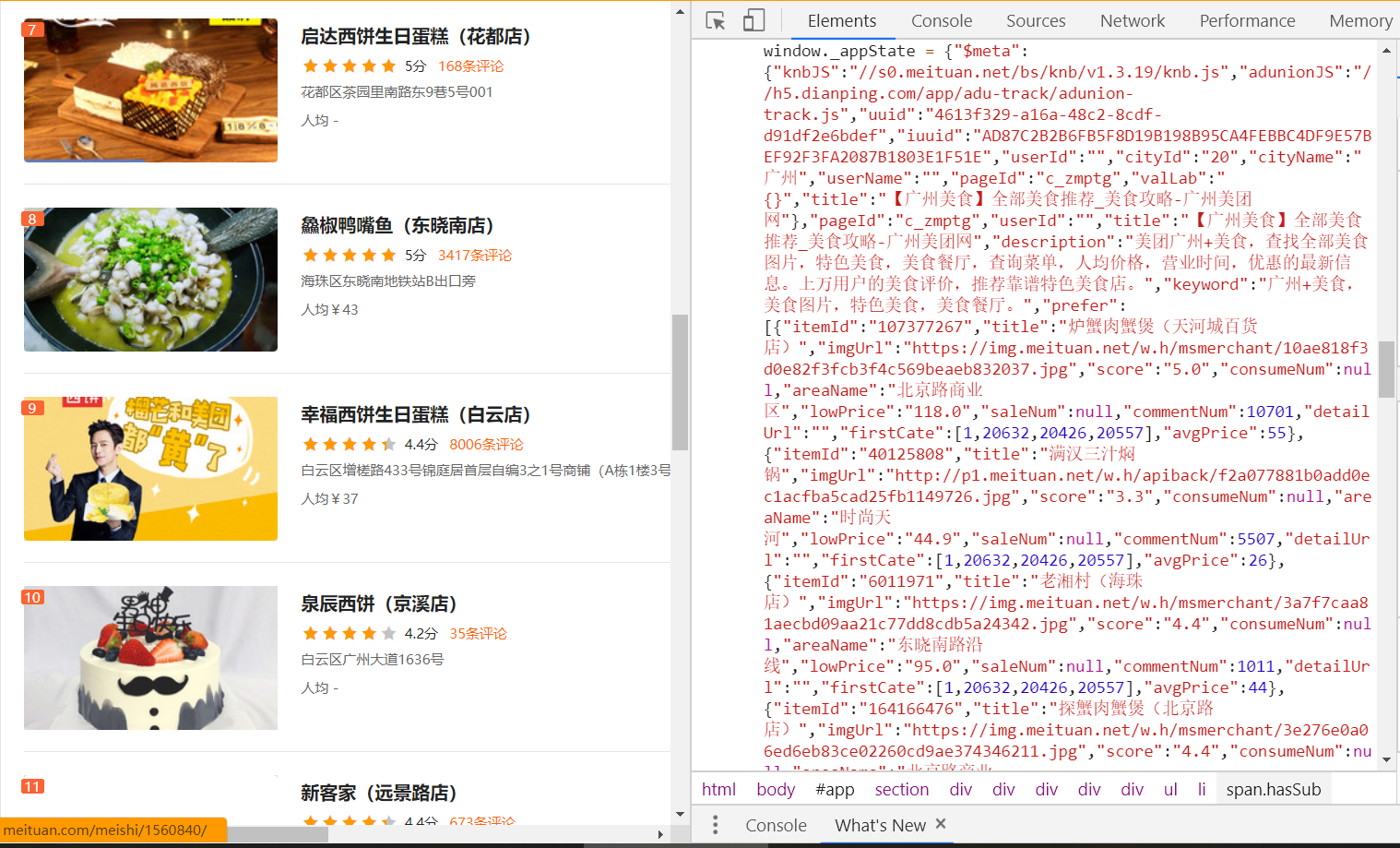
**originUrl:** http://www.meituan.com/meishi/

**riskLevel:** 1

**optimusCode: 10**

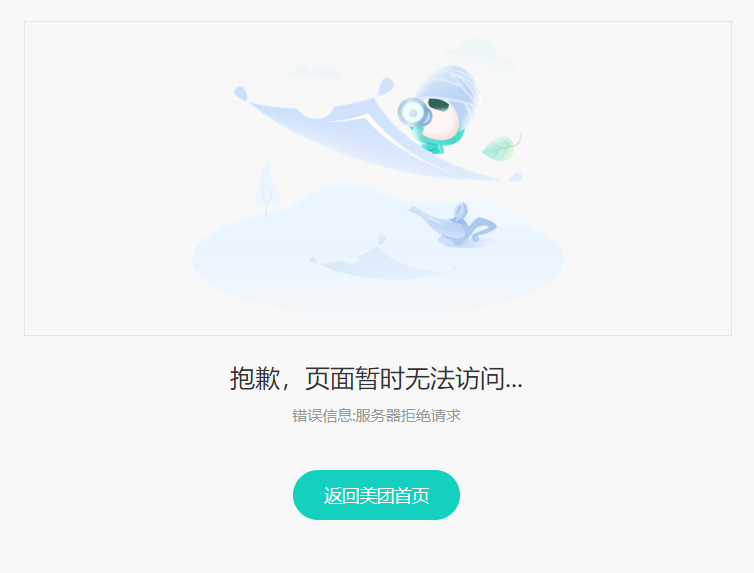
**\_token:** eJxVjstygjAUht8l2zKSEBLEna2iscYbKGrHBQoCAppyEaTTd286YxedOTP/5XyL/wvkzAc9BKEJoQLuQQ56AHVghwIFlIX8EIqwDolGuhgp4PS/Mw1DAcd8MwC9D2J2FQOjw2+xkvkDEUyVLtUPytNiaTVd3i/DJAKishQ9Va3rupMFcVl5187plqnSF1GsyglAopkjUanJU72nln+Zy82SLeLwKl0wqdPLGr7Xl/4yCtSY5aW3pu4tYa4+LGz/tYmP8VLX7H3Kp5Yx8vQwd164OZgKOtZVzGtrpFlJu24gU2/j5By+Lch9Rs9EX+ztZEhGmC+dXbwdsM9Bwl6EpkFvs eIrb9dG9gKTmD3yeUnHPBArwvjOF9a28ecWzTf9uW+/84k9uaHKvWwhe6OPabt3qspooiuOLCvjjzBVz23mzHal25pa1924oUtD90GpSAVC6QYJ2ojCaY4+6VezYpjuwfcP/VeIFA==

* + - 1. **结果**



* + - 1. **反爬虫策略分析**

请求参数必须携带加密 \_token参数，若没有该参数服务器**拒绝请求**，另外，每个\_token在一定时间内有效



* + 1. **Mobile端美食主页**
       1. **请求**
          1. **Request URL**

https://meishi.meituan.com/i/?ci=20

* + - * 1. **Request Method:** GET
        2. **Request Headers**

**Accept:** text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3

**Accept-Encoding:** gzip, deflate, br

**Accept-Language:** zh,zh-TW;q=0.9,en-US;q=0.8,en;q=0.7,zh-CN;q=0.6

**Cache-Control:** max-age=0

**Connection:** keep-alive

**Cookie:** \_\_mta=119789239.1560228713048.1561346435444.1561346469668.5; \_lxsdk\_cuid=16b34d7ff07c8-07b64189060e93-e353165-144000-16b34d7ff08c8; \_hc.v=6bd5e373-9e29-674d-16ed-e3a84a9dce7b.1559968514; iuuid=AD87C2B2B6FB5F8D19B198B95CA4FEBBC4DF9E57BEF92F3FA2087B1803E1F51E; cityname=%E5%B9%BF%E5%B7%9E; \_lxsdk=AD87C2B2B6FB5F8D19B198B95CA4FEBBC4DF9E57BEF92F3FA2087B1803E1F51E; webp=1; \_\_utmz=74597006.1560040810.1.1.utmcsr=blog.csdn.net|utmccn=(referral)|utmcmd=referral|utmcct=/xing851483876/article/details/81842329; \_\_utma=74597006.648188052.1560040810.1560228630.1560229934.3; i\_extend=C\_b0E280564580298145705312641275118984368883\_e849009944306480013\_a%e6%97%a9%e8%8c%b6Gimthomepagecategory11H\_\_a100173\_\_b3; uuid=4613f329-a16a-48c2-8cdf-d91df2e6bdef; \_lx\_utm=utm\_source%3Dblog.csdn.net%26utm\_medium%3Dreferral%26utm\_content%3D%252Fxing851483876%252Farticle%252Fdetails%252F81842329; lat=22.987182; lng=113.345815; client-id=527013c7-08b6-4475-a831-28180da22822; logan\_custom\_report=; ci=20; logan\_session\_token=jmcq8cpi400yyye7d1kw; \_lxsdk\_s=16b877fe913-043-2aa-815%7C%7C4

**Host:** meishi.meituan.com

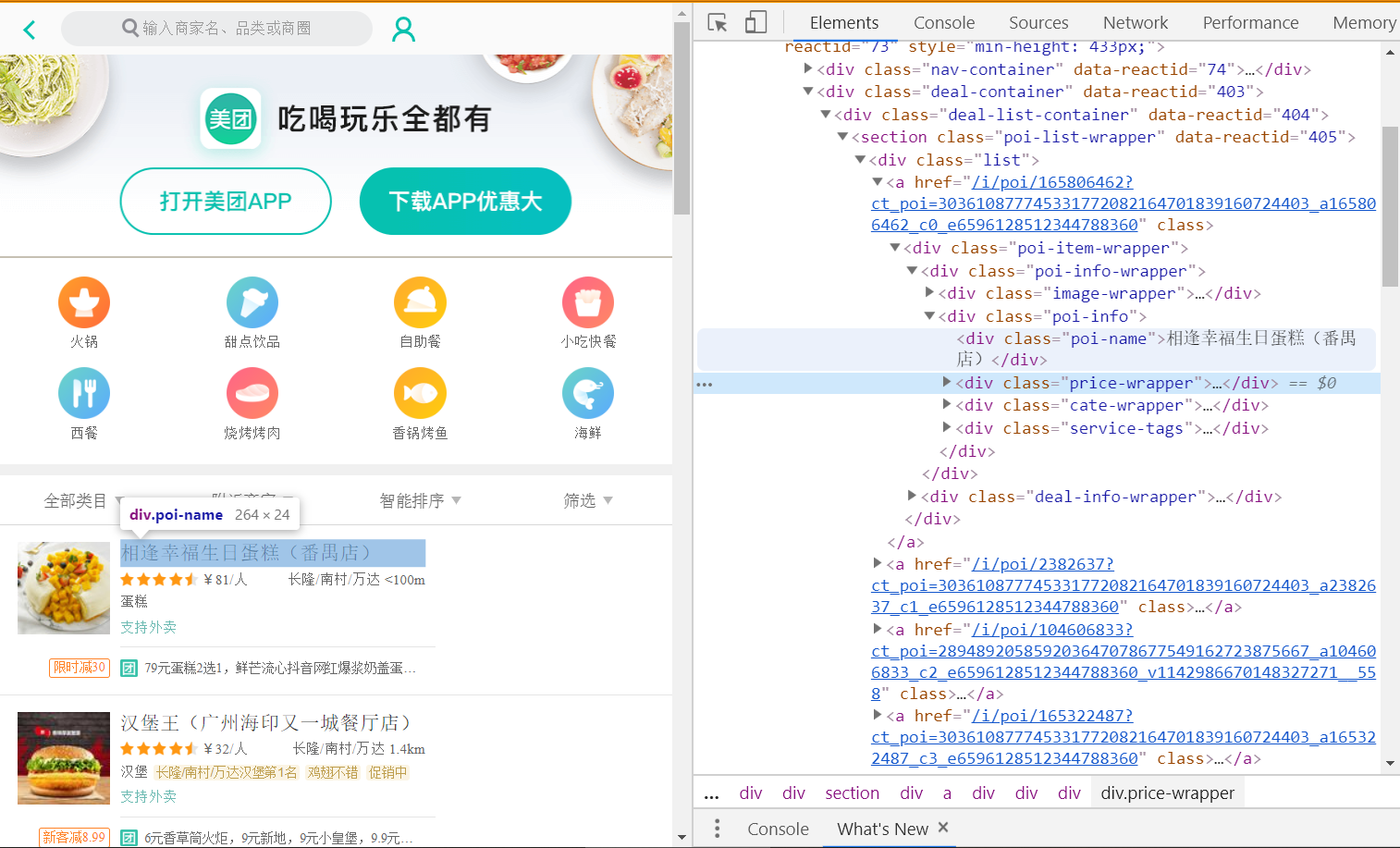
**Upgrade-Insecure-Requests:** 1

**User-Agent:** Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36

* + - * 1. **Query String Parameters**

**ci:** 20

* + - 1. **结果**

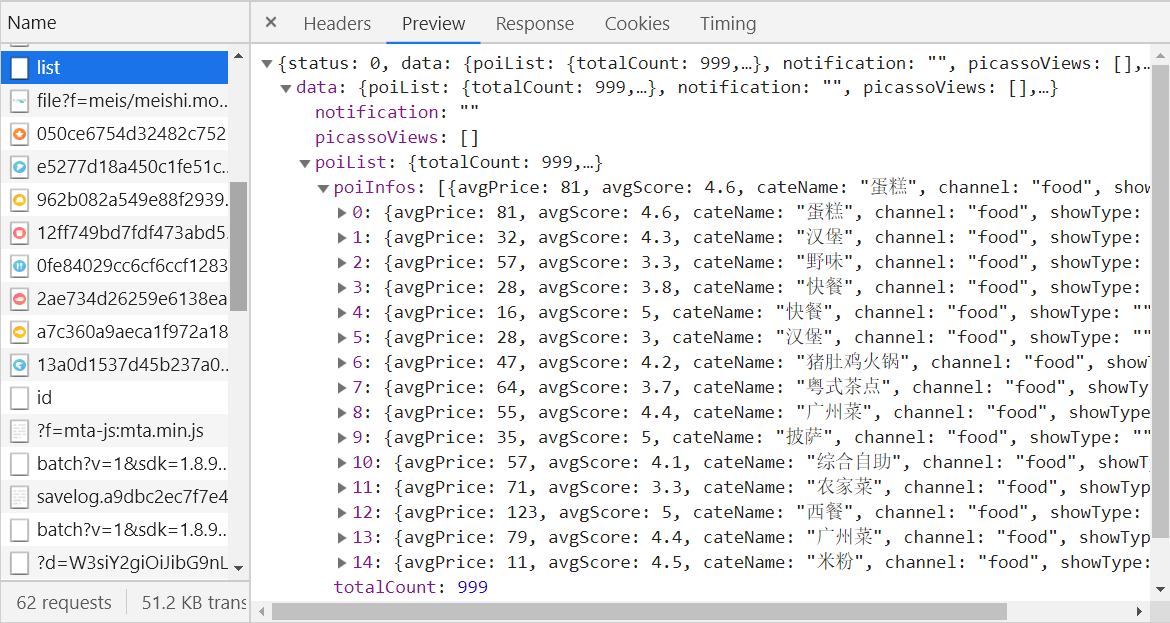


* + - 1. **反爬虫策略分析**

页面采用了客户端渲染的方式加载数据，因此如果不使用**模拟浏览器**则无法爬取到有效数据



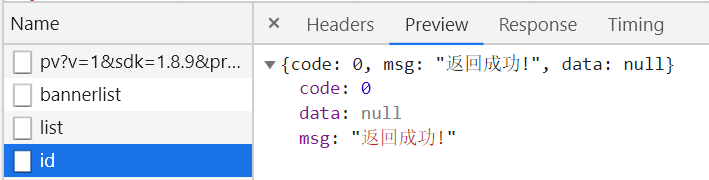
真正的XHR请求为

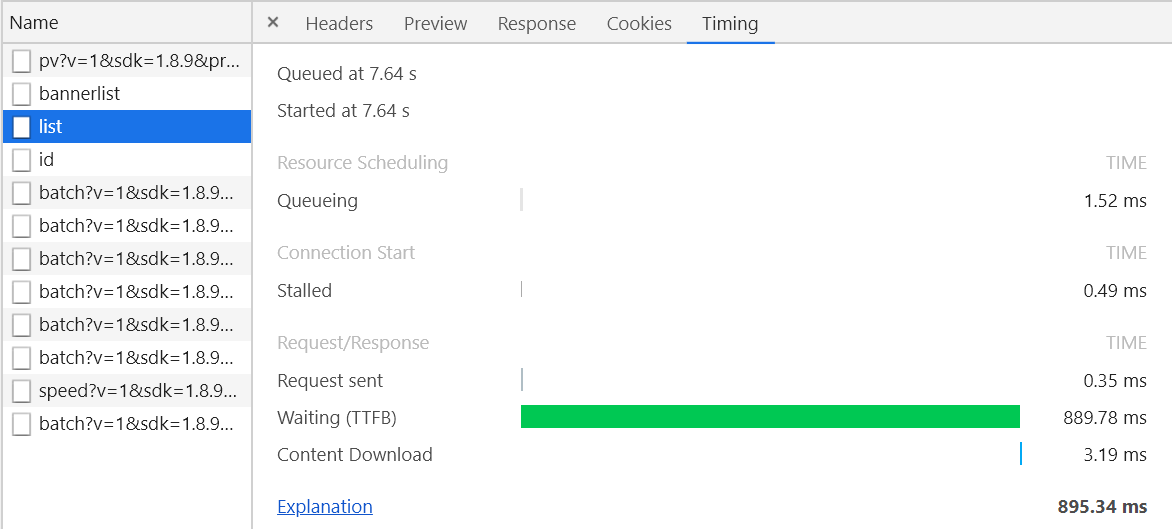


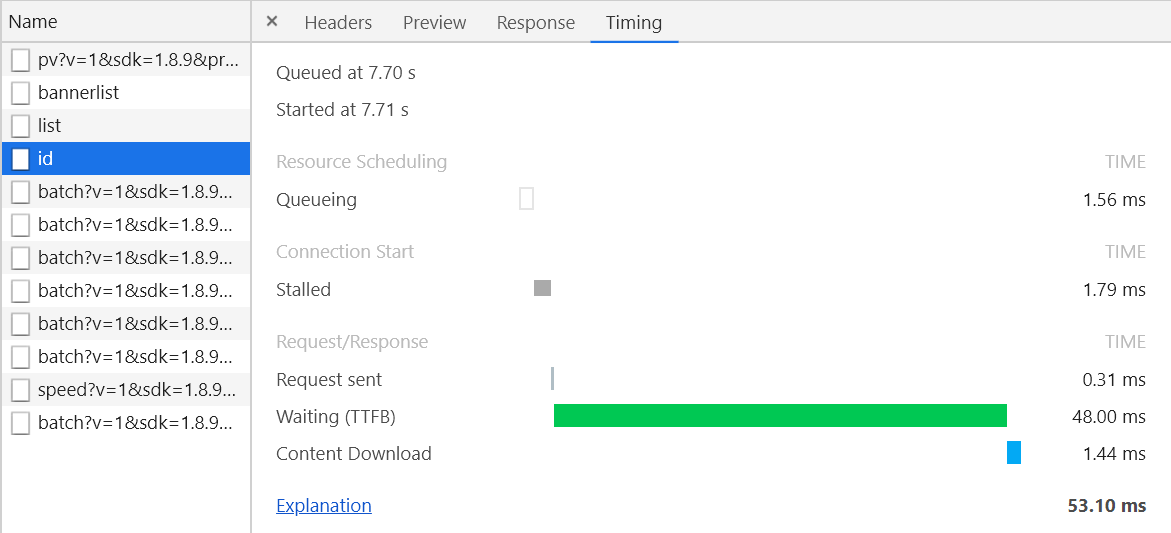
但是模拟请求后仍然得不到有效信息



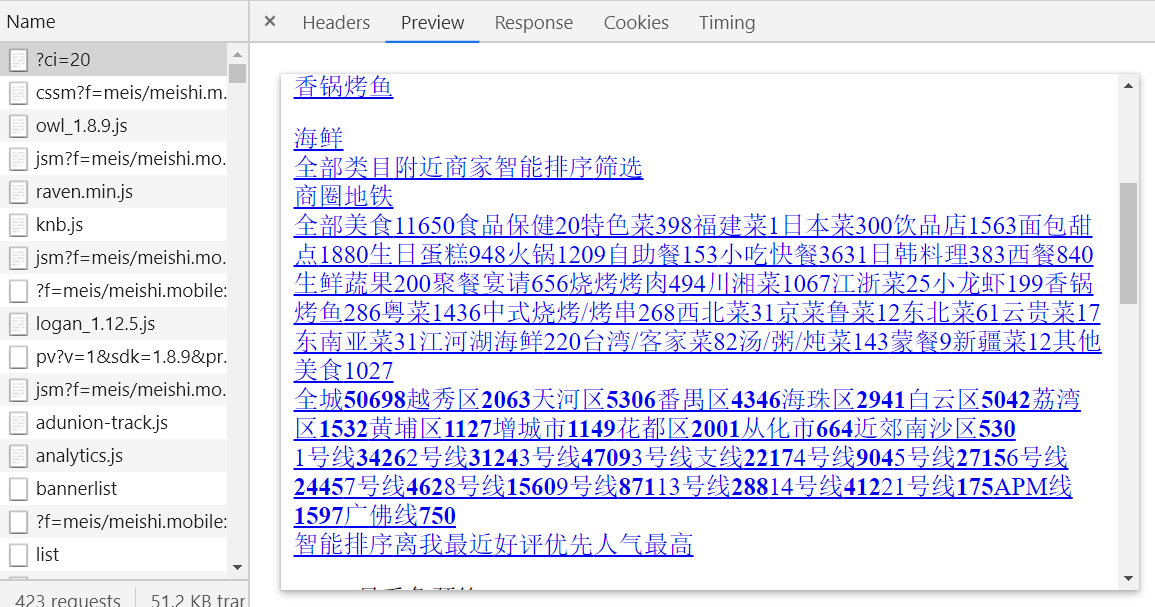
断点分析后发现原因是在发出请求list后，浏览器会等待一段时间，同时发送一个名为id的get请求，服务器接收到id请求后才会回应list请求







经过对页面其他部分分析发现页面中有其他有效数据可以利用：**美食分类**和**地域分布**



* + 1. **PC端美食关键字搜索结果页**
       1. **请求**
          1. **Request URL**

<https://apimobile.meituan.com/group/v4/poi/pcsearch/20?uuid=29d923a77824412194e3.1560221341.1.0.0&userid=-1&limit=32&offset=32&cateId=-1&q=%E6%97%A9%E8%8C%B6>

* + - * 1. **Request Method:** GET
        2. **Request Headers**

**Origin:** https://gz.meituan.com

**Referer:** https://gz.meituan.com/s/%E6%97%A9%E8%8C%B6/

**User-Agent:** Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36

* + - * 1. **Query String Parameters**

**uuid:** 29d923a77824412194e3.1560221341.1.0.0

**userid:** -1

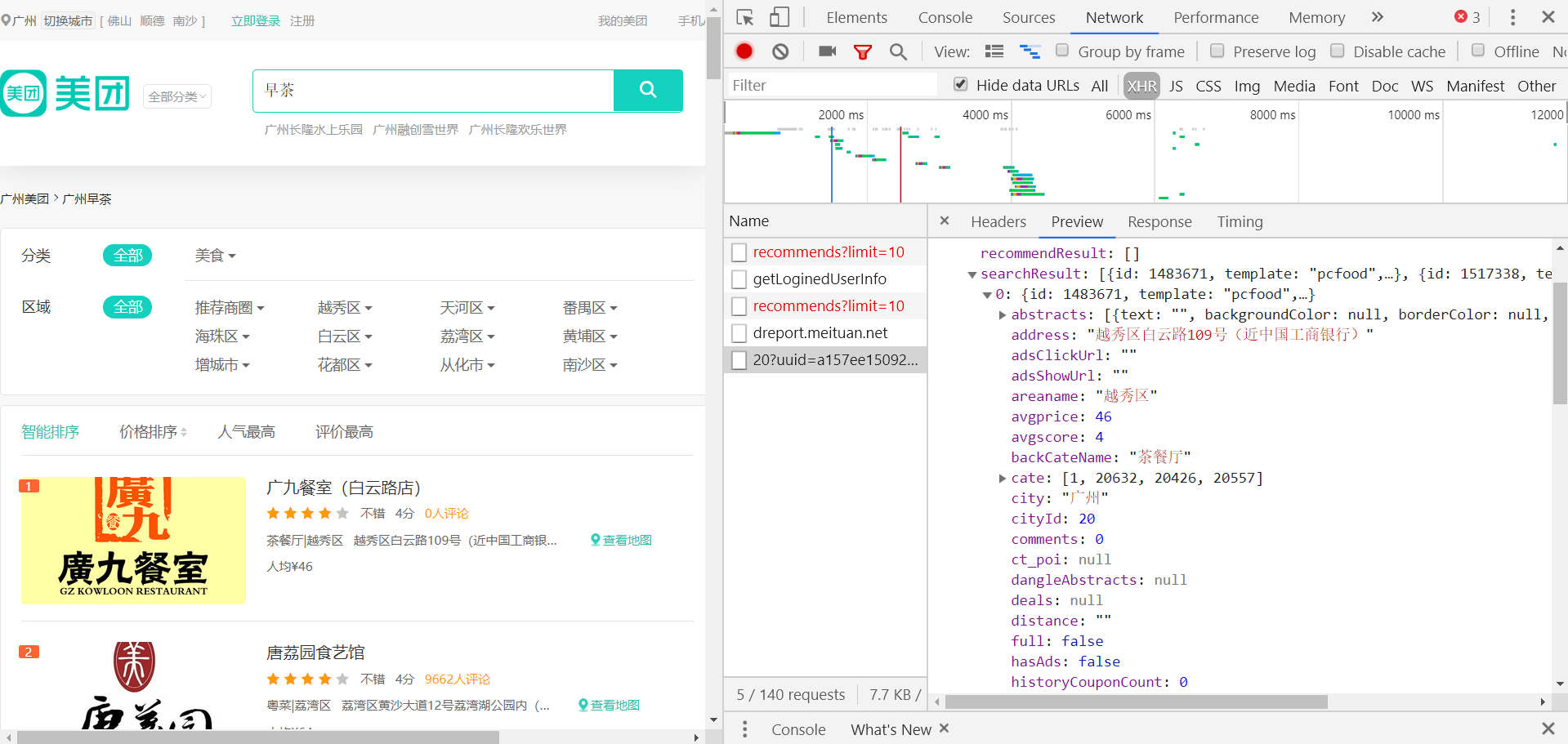
**limit:** 32

**offset:** 32

**cateId:** -1

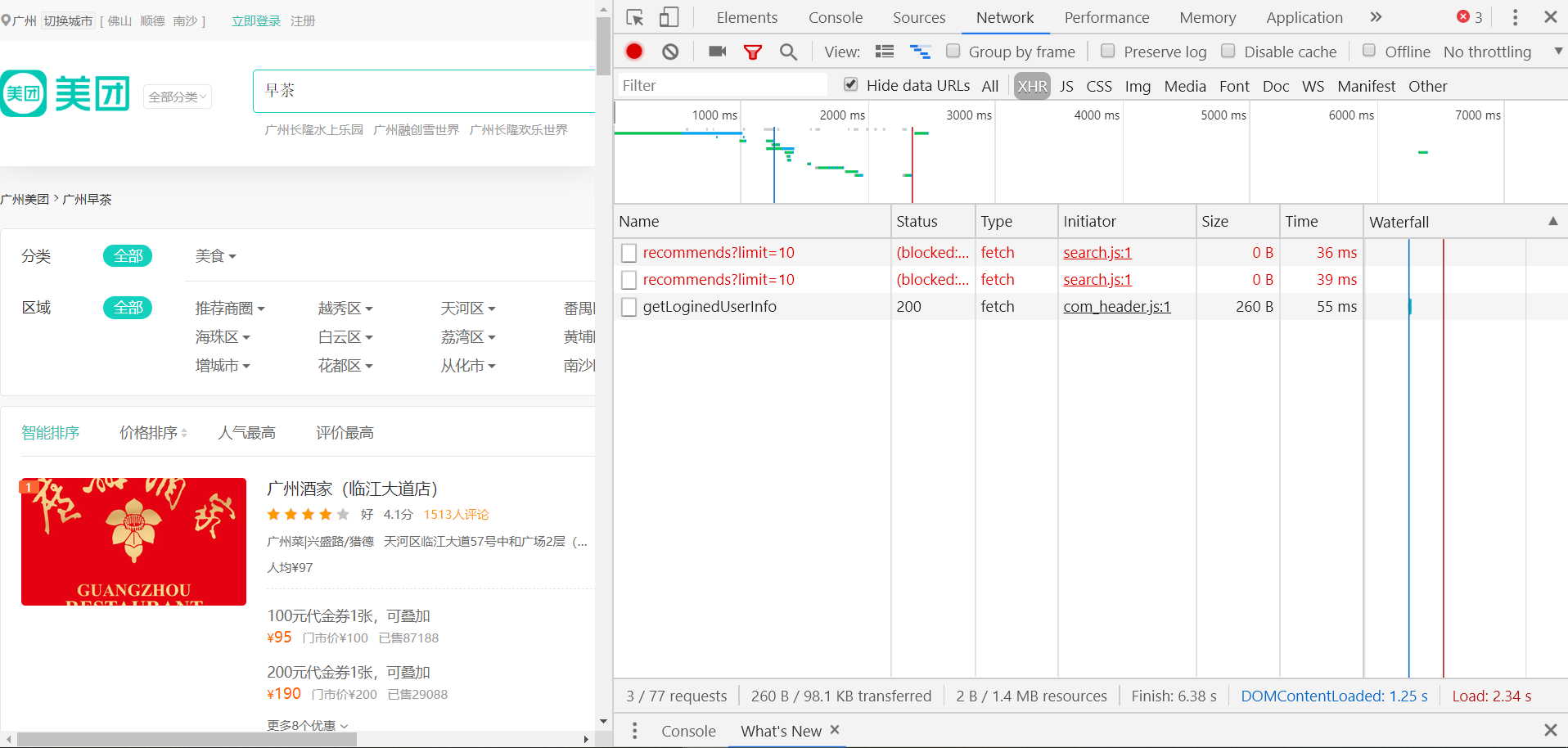
**q:** 早茶

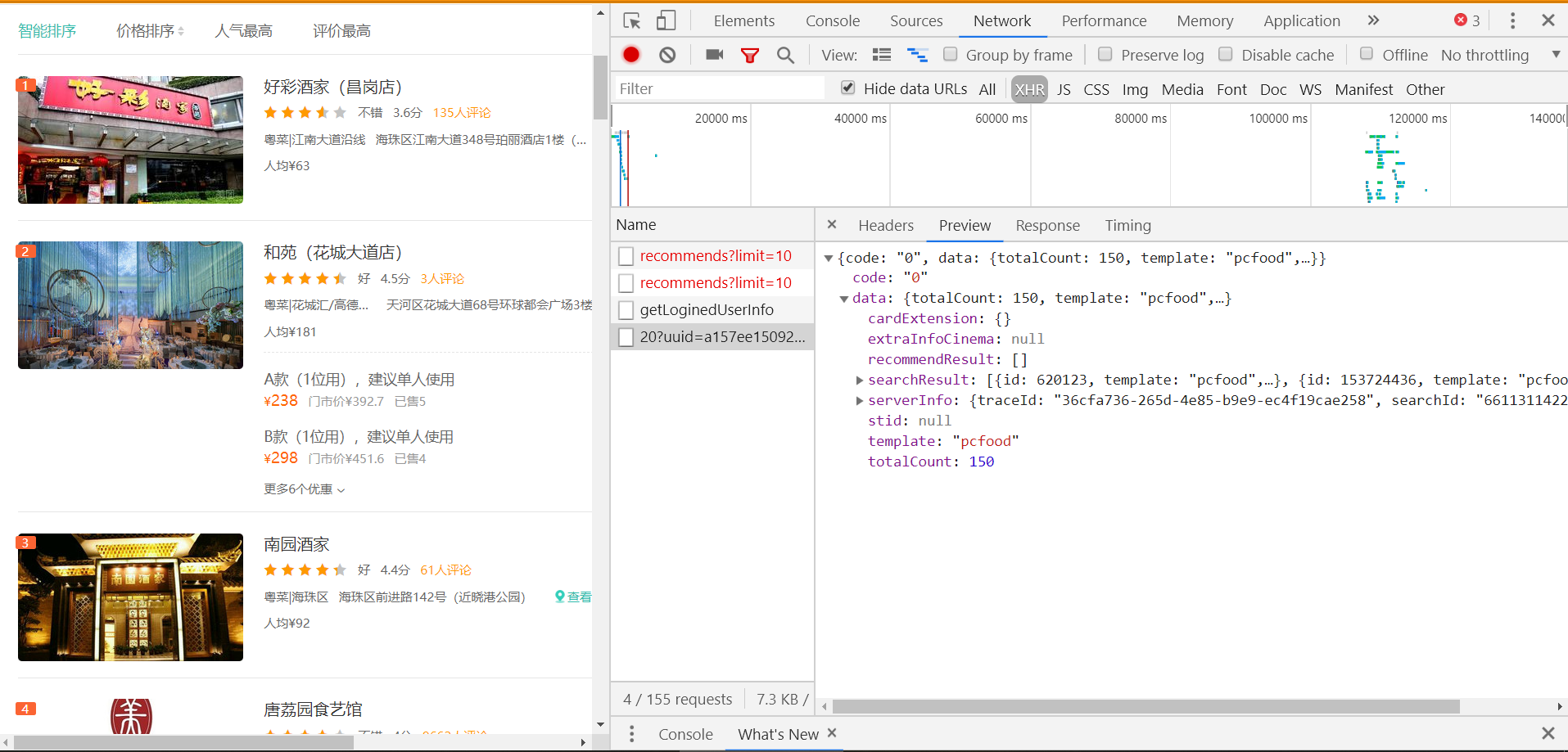
* + - 1. **结果**



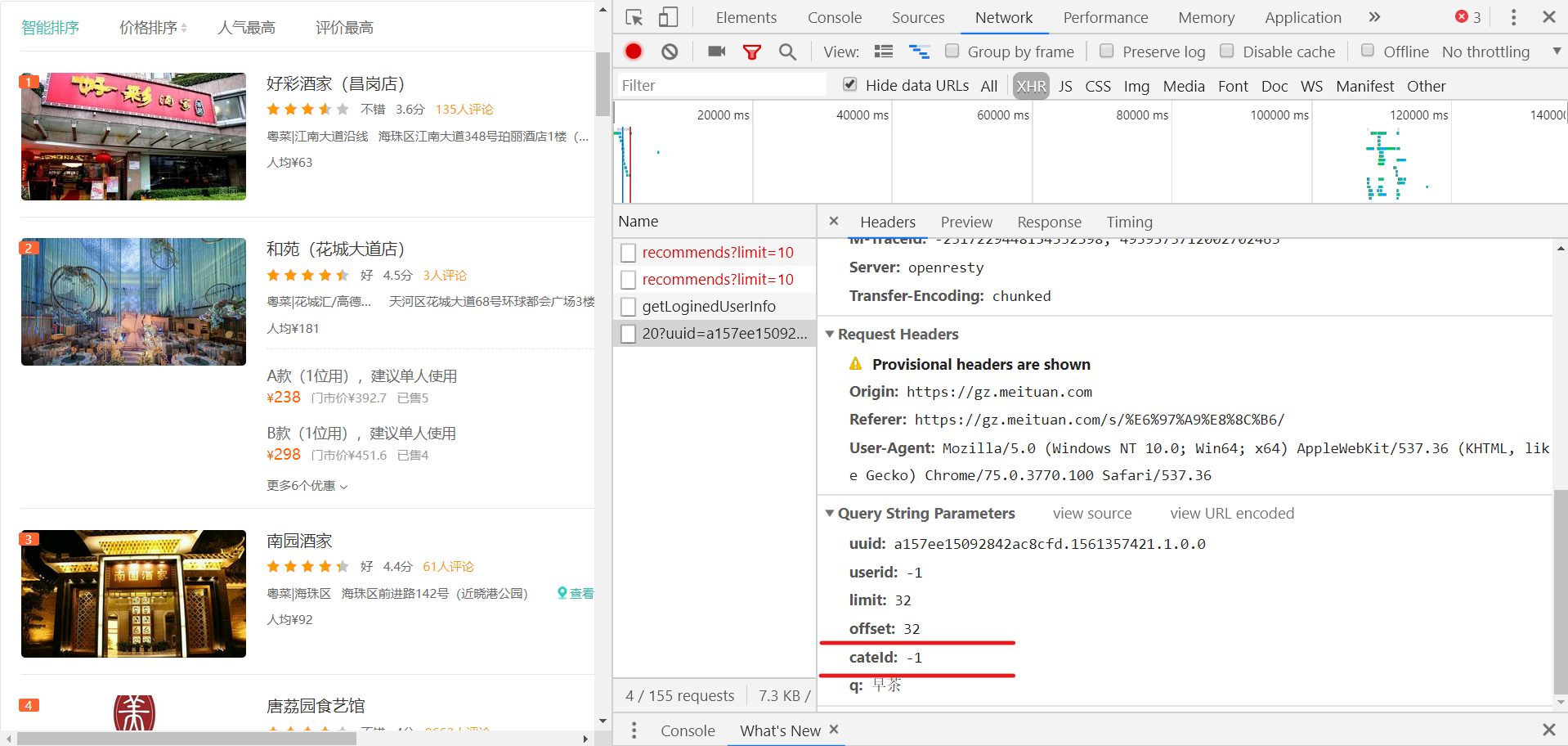
* + - 1. **反爬虫策略分析**

通过使用浏览器开发者工具分析发现搜索结果页首页是**服务器端渲染**的结果，从第二页开始是**客户端渲染**的结果





并且通过请求参数中的**limit**和**offset**设置每页显示数量和排序序号



请求过于频繁会出现验证码界面

****

* + 1. **Moblie端美食关键字搜索结果页**
       1. **请求**
          1. **Request URL**

https://i.meituan.com/s/a?cid=-1&bid=-1&sid=defaults&p=2&ciid=20&bizType=area&csp=&cevent=imt%2Fhomepage%2Fsearch&nocount=true&stid\_b=\_b2&w=%E6%97%A9%E8%8C%B6

* + - * 1. **Request Method:** GET
        2. **Request Headers**

**Accept:** text/html

**Referer:** https://i.meituan.com/s/guangzhou-%E6%97%A9%E8%8C%B6?p=2

**User-Agent:** Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36

**X-Requested-With:** XMLHttpRequest

* + - * 1. **Query String Parameters**

**cid:** -1

**bid:** -1

**sid:** defaults

**p:** 2

**ciid:** 20

**bizType:** area

**csp:**

**cevent:** imt/homepage/search

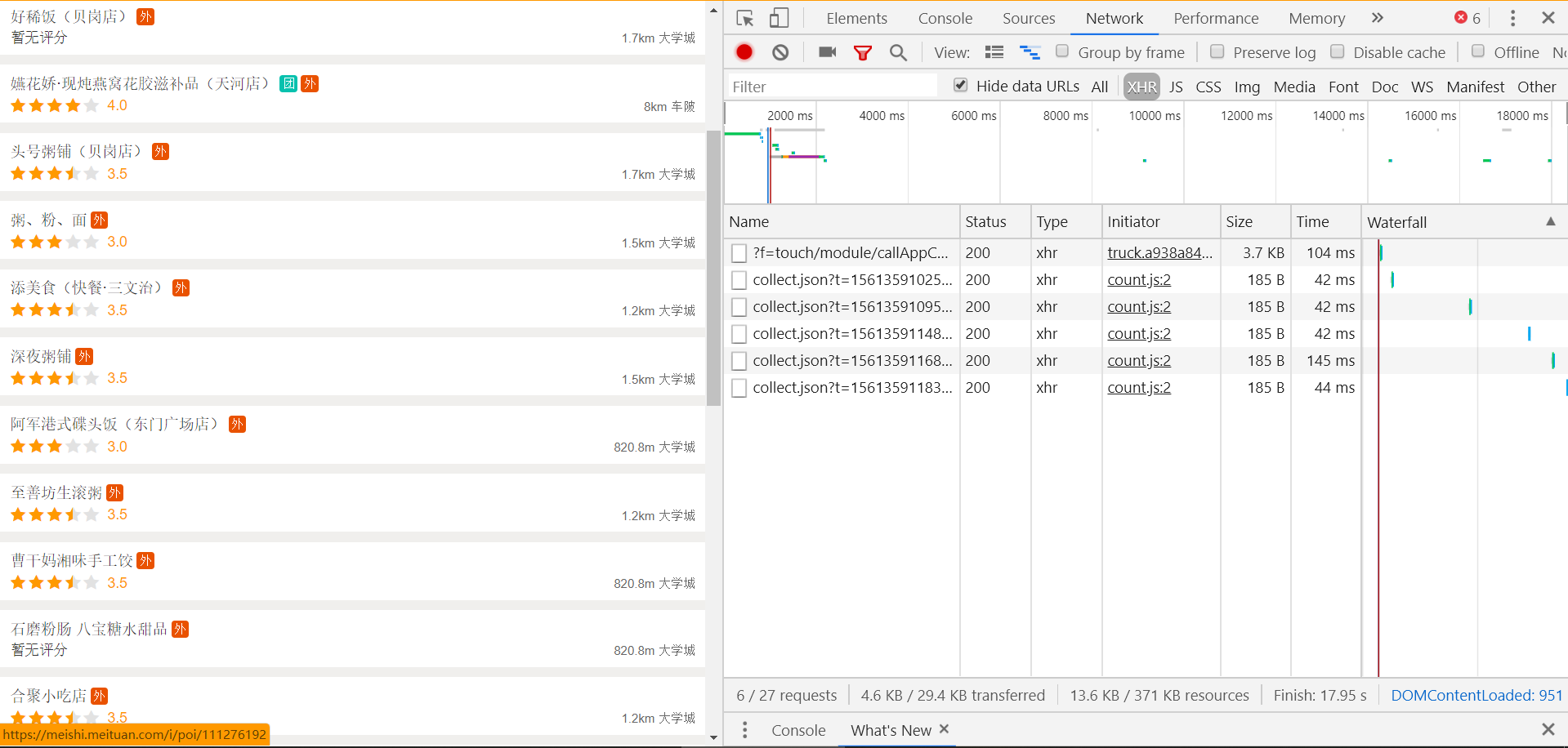
**nocount:** true

**stid\_b:** \_b2

**w:** 早茶

* + - 1. **结果**

搜索结果大多数为**外卖**



* + - 1. **反爬虫策略分析**

同样是第一页服务器端渲染，下一页开始客户端渲染

此页面无太多有用信息，不做爬虫。

* + 1. **PC端美食店铺主页请求**
       1. **请求**
          1. **Request URL**

<http://www.meituan.com/meishi/165806462/>

* + - * 1. **Request Method:** GET
        2. **Request Headers**

**Accept:** text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3

**Accept-Encoding:** gzip, deflate

**Accept-Language:** zh,en-US;q=0.9,en;q=0.8

**Connection:** keep-alive

**Cookie:** \_lxsdk\_cuid=16b368bea30c8-01d455e881d3b3-2e5b0b4d-144000-16b368bea31c8; client-id=e5346ef7-123d-4599-83e4-98bb6e8dcf98; \_hc.v=5be0bf46-d5f9-941e-efab-91a9238cec08.1559988540; iuuid=FB2257B9F6C3900FF80AB5034F1580F3C2BAF47B0E2CC32EFF71133A2C36E18F; cityname=%E5%B9%BF%E5%B7%9E; \_lxsdk=FB2257B9F6C3900FF80AB5034F1580F3C2BAF47B0E2CC32EFF71133A2C36E18F; ci=20; rvct=20%2C1%2C92; \_ga=GA1.2.594880113.1560266594; \_gid=GA1.2.1565384567.1560266594; uuid=089dfafe6db147008b39.1560302590.1.0.0; lat=22.987182; lng=113.345815; \_lxsdk\_s=16b49988747-9b0-5d7-3eb%7C%7C9

**Host:** www.meituan.com

**Sec-Fetch-Mode:** navigate

**Sec-Fetch-Site:** same-site

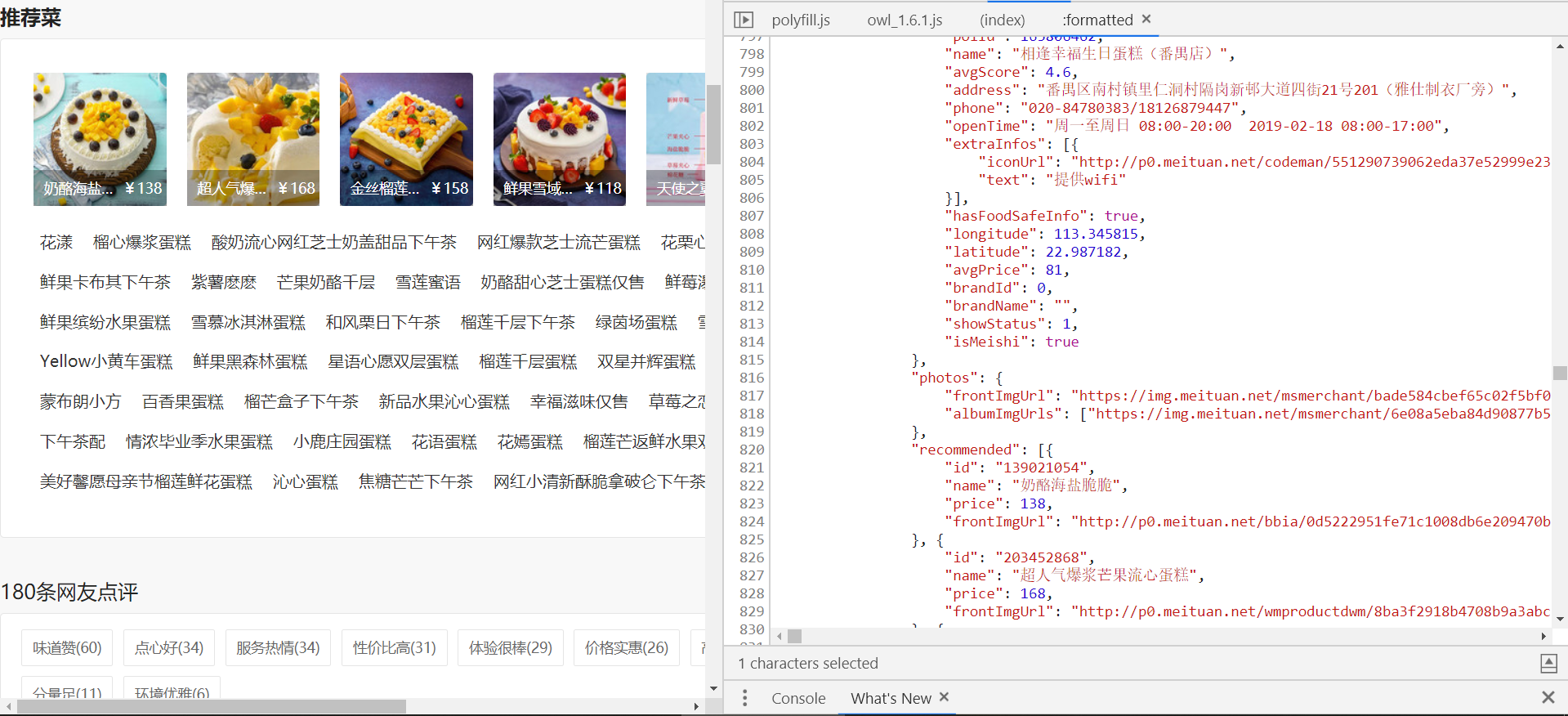
**Sec-Fetch-User:** ?1

**Upgrade-Insecure-Requests:** 1

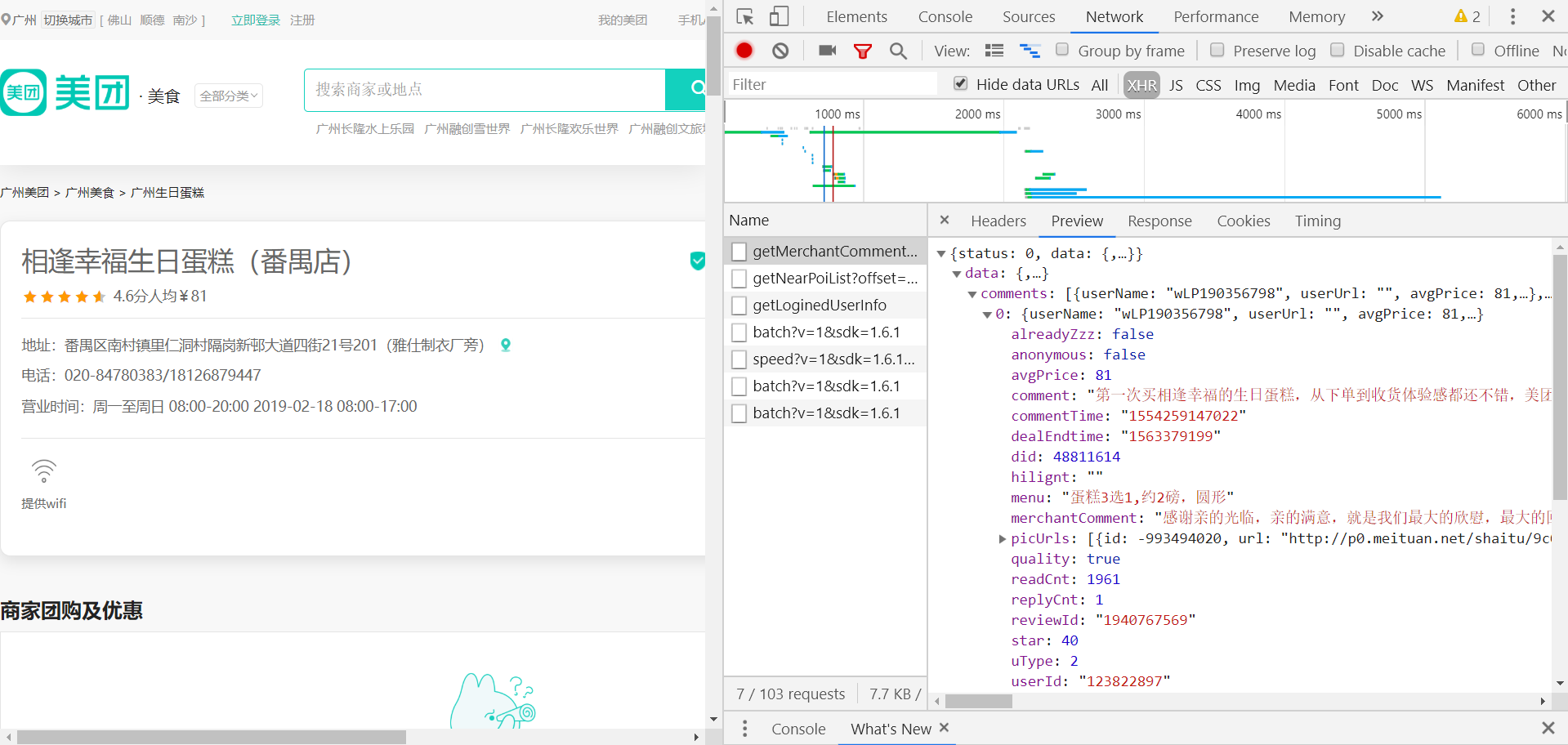
**User-Agent:** Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36

* + - 1. **结果**

解析页面可以获得店铺信息和店铺推荐菜等信息

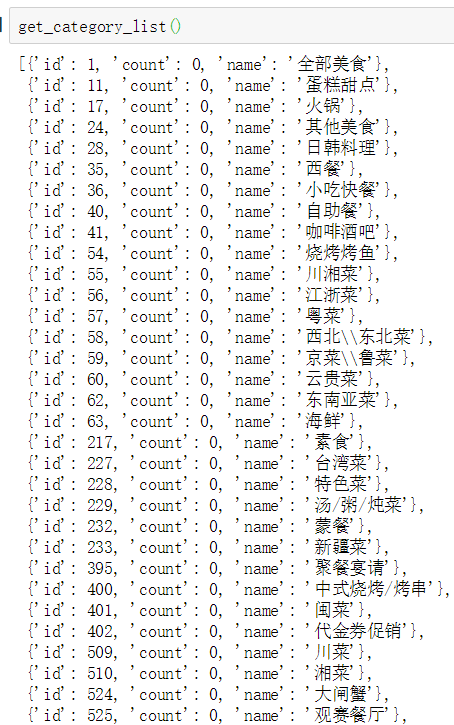


通过开发者工具可以获得该店铺的**评论**的XHR



* + - 1. **反爬虫策略分析**

由于评论等进一步信息是由**XHR异步获取**的，直接爬取该页面只能获取到有限的信息。另外，由于Moblie端的美食店铺主页的信息与2.3.4中提到的两个XHR相结合一样，因此不具体展开。



* + 1. **PC端店铺评论请求**
       1. **请求**
          1. **Request URL**

<https://www.meituan.com/meishi/api/poi/getMerchantComment?uuid=1f43caa5-eb7d-46bc-ae31-9485eb28d9dd&platform=1&partner=126&originUrl=https%3A%2F%2Fwww.meituan.com%2Fmeishi%2F5548637%2F&riskLevel=1&optimusCode=1&id=5548637&userId=&offset=0&pageSize=10&sortType=1>

* + - * 1. **Request Method:** GET
        2. **Request Headers**

**Accept:** application/json

**Accept-Encoding:** gzip, deflate

**Accept-Language:** zh,zh-TW;q=0.9,en-US;q=0.8,en;q=0.7,zh-CN;q=0.6

**Connection:** keep-alive

**Host:** www.meituan.com

**Referer:** https://www.meituan.com/meishi/5548637/

**User-Agent:** Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36

* + - * 1. **Query String Parameters**

**uuid:** 1f43caa5-eb7d-46bc-ae31-9485eb28d9dd

**platform:** 1

**partner:** 126

**originUrl:** https://www.meituan.com/meishi/5548637/

**riskLevel:** 1

**optimusCode:** 1

**id:** 5548637

**userId:**

**offset:** 0

**pageSize:** 10

**sortType:** 1

* + - 1. **结果**



* + - 1. **反爬虫策略分析**

缺少**请求头部的Host属性**有时会被美团服务器检查出为爬虫

* + 1. **Moblie端店铺评论请求**
       1. **请求**
          1. **Request URL**

<https://i.meituan.com/poi/619195/feedbacks/page_2>

* + - * 1. **Request Method:** GET
        2. **Request Headers**

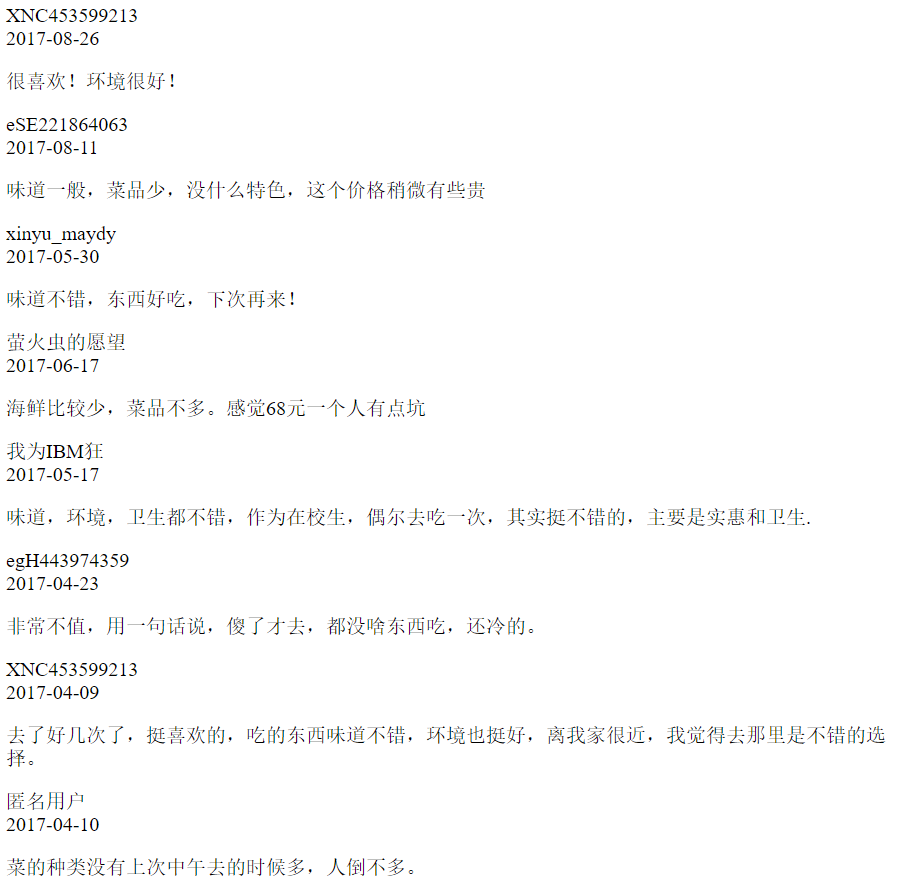
**Accept:** \*/\*

**Referer:** https://i.meituan.com/poi/619195/feedbacks

**User-Agent:** Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36

**X-Requested-With:** XMLHttpRequest

* + - 1. **结果**



* + - 1. **反爬虫策略分析**

信息量不大，HTML处理麻烦，不对此页面进行爬虫

* + 1. **Mobile端店铺推荐菜请求**
       1. **请求**
          1. **Request URL**

<https://meishi.meituan.com/i/api/dish/poi>

* + - * 1. **Request Method:** POST
        2. **Request Headers**

**Accept:** application/json

**Content-Type:** application/json

**Origin:** https://meishi.meituan.com

**Referer:** https://meishi.meituan.com/i/poi/619195?ct\_poi=131212051330676446061144842076618169785\_e7169842945503864939\_a%25e6%2597%25a9%25e8%258c%25b6&cevent=imt%2Fpoilist%2Fitem

**User-Agent:** Mozilla/5.0 (Linux; Android 5.0; SM-G900P Build/LRX21T) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Mobile Safari/537.36

**x-requested-with: XMLHttpRequest**

* + - * 1. **Request Payload**

**app: ""**

**optimusCode:** 10

**originUrl:** "http://meishi.meituan.com/i/poi/619195?ct\_poi=131212051330676446061144842076618169785\_e7169842945503864939\_a%25e6%2597%25a9%25e8%258c%25b6&cevent=imt%2Fpoilist%2Fitem"

**partner:** 126

**platform:** 3

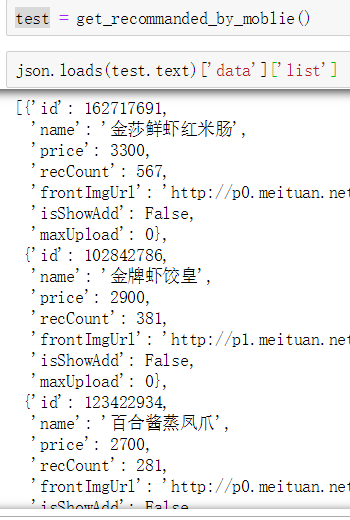
**poiId:** 619195

**riskLevel:** 1

**uuid:** "1f43caa5-eb7d-46bc-ae31-9485eb28d9dd"

**version:** "8.3.3"

* + - 1. **结果**



* 1. **总结反爬虫策略以及可执行的对应反反爬虫策略**
     1. **请求参数加密**
        1. **概述**

请求参数中加入通过 js 的加密库生成动态的 token，同时加密库再进行混淆。

* + - 1. **应对方法**

1. 爬取手机端页面绕过token加密；
2. 使用浏览器开发者工具慢慢调试，找到加密原理；
3. 使用模拟浏览器方式模拟用户行为；
4. 每个token有一定有效时间，手动截取一个token使用，待其过期后再重新获取。
   * 1. **有效数据异步获取**
        1. **概述**

网页中的数据不由服务器渲染完成，而是交给客户端异步获取数据再进行渲染，因此若是直接对原始网页进行爬虫无法爬取到有效数据。

* + - 1. **应对方法**

1. 使用浏览器开发者工具中Network选项查看XHR请求进行模拟；
2. 多查看同一级别不同网页进行比较，必要时使用开发者工具断点调试进行分析；
3. 使用模拟浏览器设置等待时间或者等待目标element加载完成再进行数据获取；
4. 使用XHR进行爬虫时，尽可能模拟请求头可以绕过反爬虫检测，特别是User-Agent属性和Host属性。
   * 1. **多个XHR结合防御爬虫**
        1. **概述**

即使找到XHR，但是经过爬虫并不能获取到有效的数据，原因是服务器端进行了多个XHR互相验证的反爬虫方法。

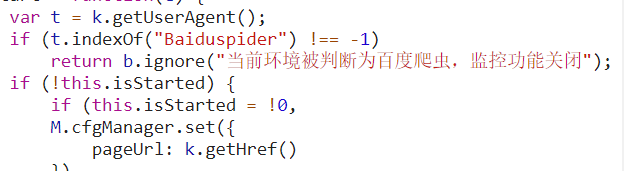
* + - 1. **应对方法**

1. 使用模拟浏览器模拟用户行为进行爬虫；
2. 此反爬虫策略美团多用于手机端页面，可更换为PC端页面进行爬虫。
   * 1. **要求验证码**
        1. **概述**

服务器对于短时间访问页面过于频繁的用户判定为有爬虫的嫌疑，要求进行验证码进行人工验证。

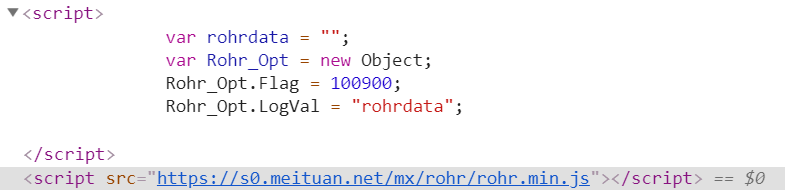
* + - 1. **应对方法**

1. 每次爬取数据后sleep随机时间，降低爬取速率；
2. 采用IP池代理方式，降低单个IP的访问次数；
3. 每次验证通过都可以维持一段时间不出现验证码，可以间歇性手动进行验证码验证；
4. 接入打码平台进行验证码识别；
5. 机器学习；
6. 手机端页面验证码出现次数较少，可以爬取手机端页面绕过验证码识别；
7. 尽可能找到XHR代替整个网页的爬取，可以有效避开验证码；
8. 模拟搜索引擎爬虫



1. **本文的方法**
   1. **分析js文件找到token的加密算法**

用浏览器的开发者工具调试，可以发现token加密是由一个rohr.min.js的文件控制的



查看文件中的变量有token以及\_token字段



根据reload函数的触发，对应分析其中参数及函数

rId: 100900,

ver: "1.0.6",

ts: new Date().getTime(),

cts: new Date().getTime(),

brVD: iN(),//浏览器窗口大小之类的

brR: iM(),//[[1536, 864],[1536, 834],24,24]屏幕大小和颜色

bI: iL(),//referrer和上一页

mT: [],

kT: [],

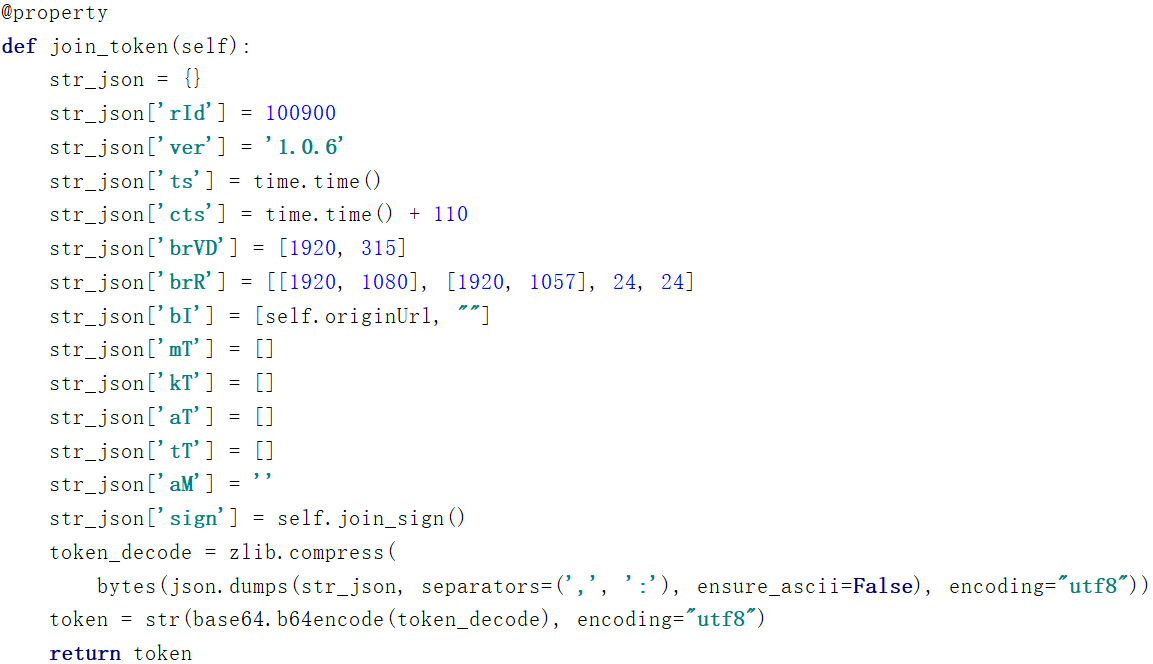
aT: [],

tT: [],

aM: iK()

加密原理为：token是先做的gzip压缩，再base64

破解的加密算法如下：



* 1. **寻找有用XHR**

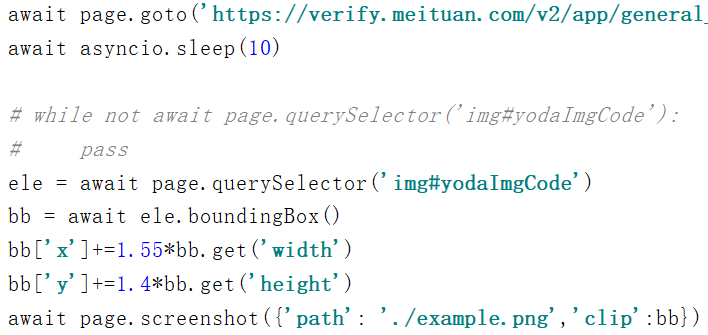
1. 对每个页面的PC端页面和Mobile页面都进行分析，查看其中的请求，找到目标数据最适合有效的爬虫方法；
2. 对于美食搜索结果页面选用PC端页面，点击下一页后发现美食列表请求为XHR于是模拟请求获取数据，修改请求中的offset和limit参数以获取完整数据（第一页的数据），并且一次获取更多数据；
3. 对于店铺评论部分数据，Mobile端为服务器端渲染，因此获取的数据为HTML格式，数据信息有限并且处理起来不如json格式方便，因此使用PC页面发现的XHR进行数据的获取，修改请求参数中的offset和pagesize参数以一次性获取更多数据。



1. 对于店铺推荐菜部分数据，发现PC端为服务器端渲染而Mobile端是使用XHR，并且XHR方法获取到的数据更为充实，选择使用Mobile端的XHR请求爬取数据。
   1. **使用打码平台破解验证码**

通过分析request是否经过转发判断是否出现验证码界面

经过源码分析发现美团的验证码图片请求生成函数较为复杂，无法破解，因此尝试使用模拟服务器获得验证码图片的src，但是发现图片为临时链接，无法二次获取，因此使用模拟浏览器获取页面后等待图片加载完成后进行截图处理，代码如下：



截图结果：



识别验证码部分首先考虑Google的tesseract库进行识别，但是没有效果

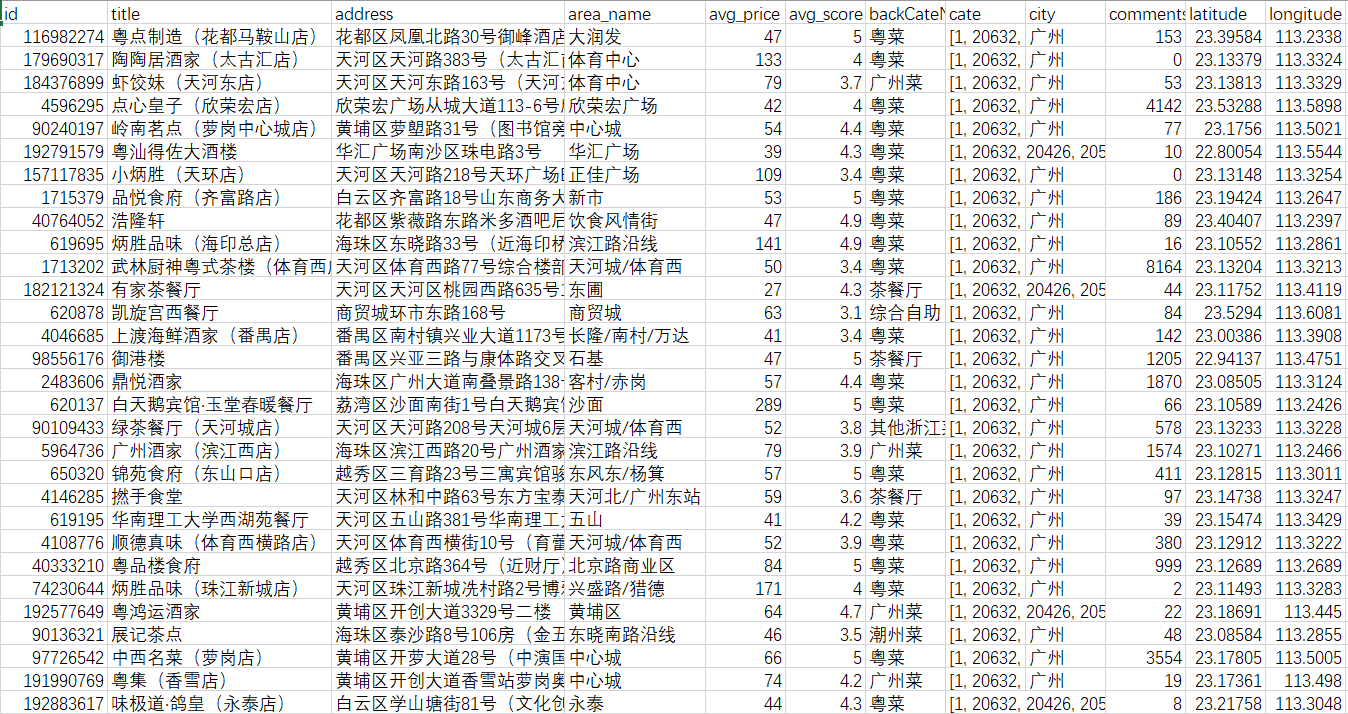
于是接入网上打码平台——斐斐打码，可以正确识别通过

* 1. **数据提取方法**

1. HTML的代码使用正则表达式匹配有用数据
2. Json数据使用python的json库loads函数整理成dict
3. get请求参数使用urllib库进行拼接
4. post请求的body部分使用json库的dumps转化成json格式
5. 数据储存使用pandas写入csv文件
6. **实验结果**
   1. **美食列表的爬取**

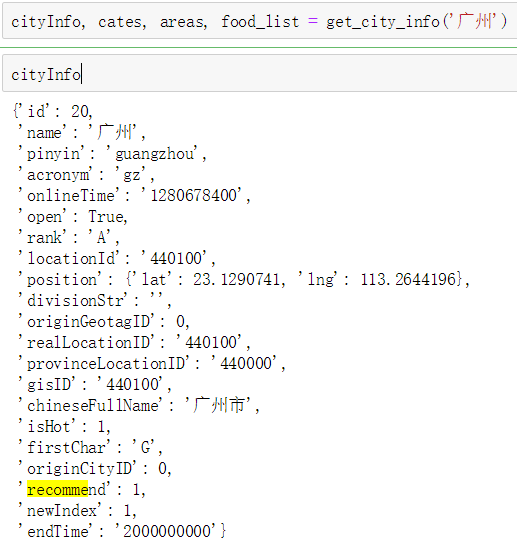


* 1. **美食搜索结果的爬取**



* 1. **城市信息的爬取**

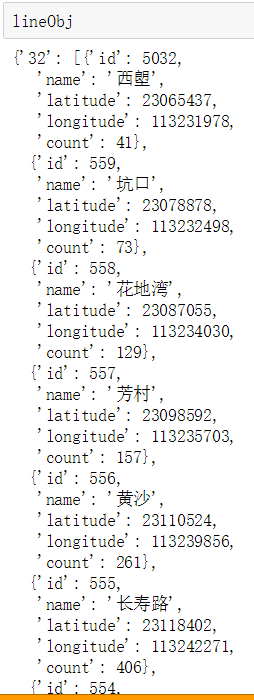
城市的基本信息



美食分类



地铁线路美食分布情况





* 1. **店铺评论的爬取**



* 1. **店铺推荐菜的爬取**



1. **总结**

美团美食部分网站总体来说爬取难度中等，主要使用了验证码、加密token、XHR异步加载、多重XHR、临时文件等策略。我们首先考虑到从PC端页面和Mobile端页面两个方向入手，对他们使用浏览器调试、请求模拟等方法分析了请求头，页面结构，数据加载方式等，再进行逐个破解。主要探索了以下的反反爬虫策略：

模拟浏览器，此方法获取数据速度较慢，资源占用较高，但操作较为简单，并且由于其与用户正常访问行为高度相似，因此可以有效绕过反爬虫。

XHR，美团的页面基本上都是首页服务器端渲染，第二页开始只请求数据进行客户端渲染。通过分析第二页的XHR的请求获取数据。

手机端对爬虫总体来说比较放松，当PC端数据过难提取时，改用手机端页面进行爬取。

对于验证码采用打码平台接入方式。

**参考文献**

[1] ×××（五号字）