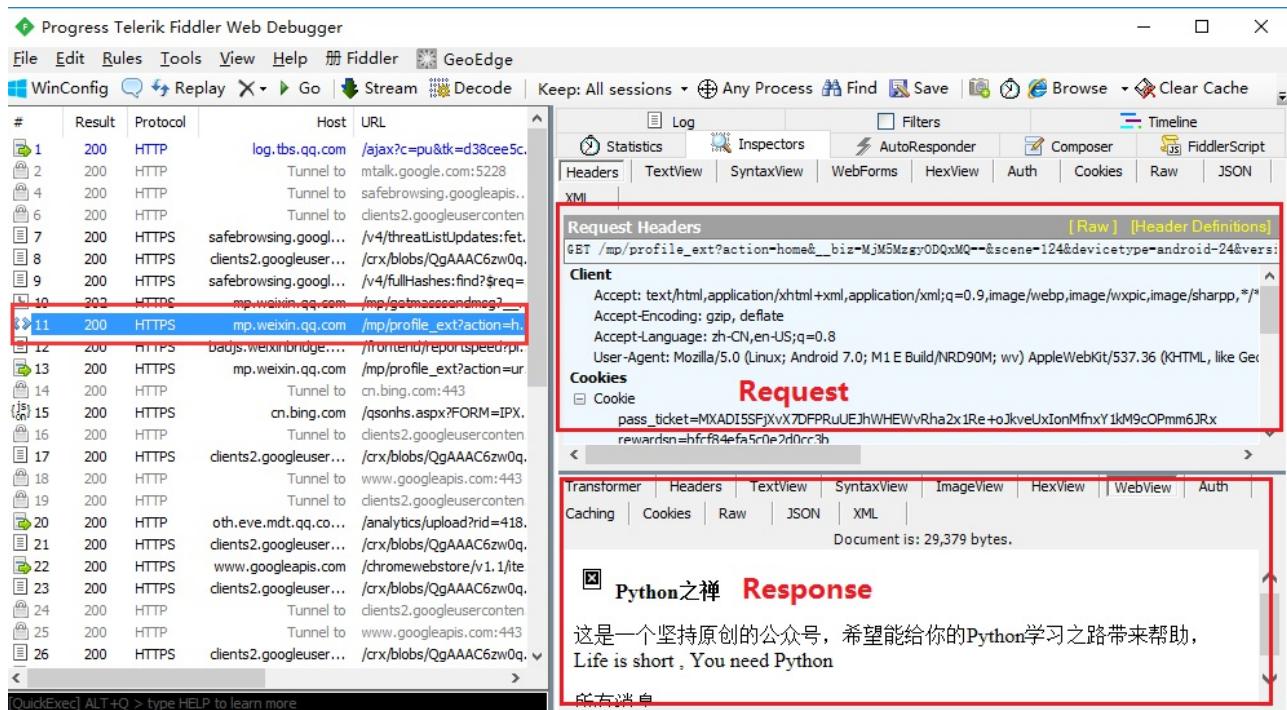


抓取第一篇微信公众号文章

上一节我们熟悉了 Fiddler 的基本操作以及每个模块所代表的意义，这节我们详细了解获取微信公众号历史文章接口的请求情况，以及如何使用 Python 模拟微信发送请求获取公众号文章的基本信息。

打开微信历史消息页面，我们从 Fiddler 看到了很多请求，为了找到微信历史文章的接口，我们要逐个查看 Response 返回的内容，最后发现第 11 个请求

"https://mp.weixin.qq.com/mp/profile_ext?action=home..."就是我们要寻找的（我是怎么找到的呢？这个和你的经验有关，你可以点击逐个请求，看看返回的Response内容是不是期望的内容）

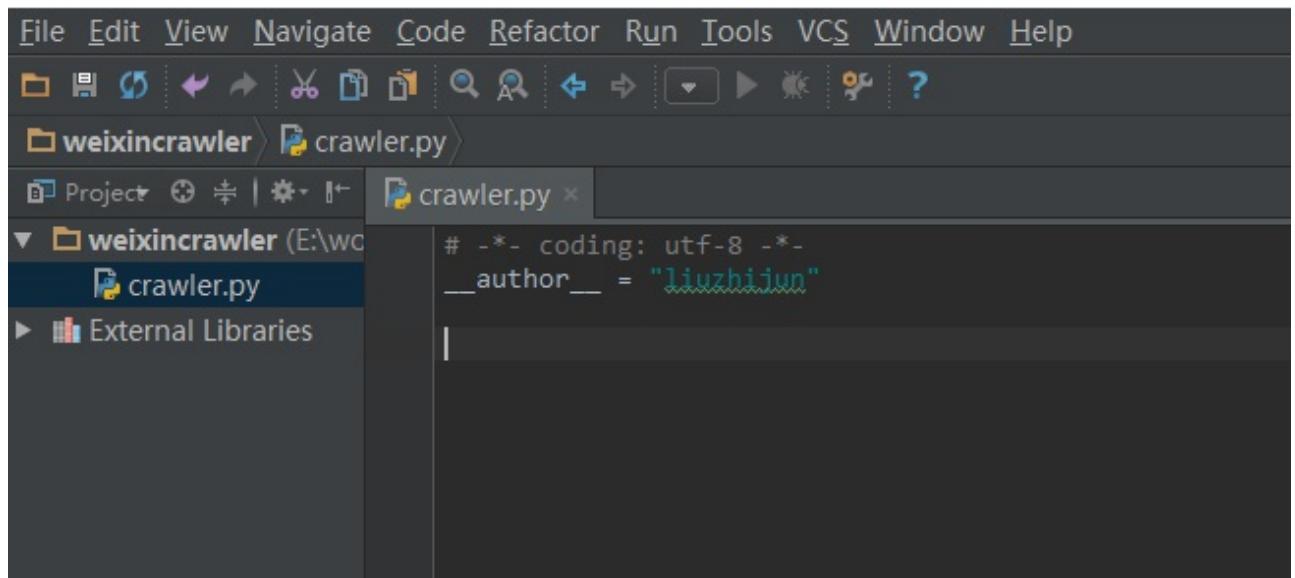


确定微信公众号的请求HOST是 `mp.weixin.qq.com` 之后，我们可以使用过滤器来过滤掉不相关的请求。

爬虫的基本原理就是模拟浏览器发送 HTTP 请求，然后从服务器得到响应结果，现在我们就用 Python 实现如何发送一个 HTTP 请求。这里我们使用 requests 库来发送请求。

创建一个 Pycharm 项目

我们使用 Pycharm 作为开发工具，你也可以使用其它你熟悉的工具，Python 环境是 Python3（推荐使用 Python3.6），先创建一个项目 weixincrawler

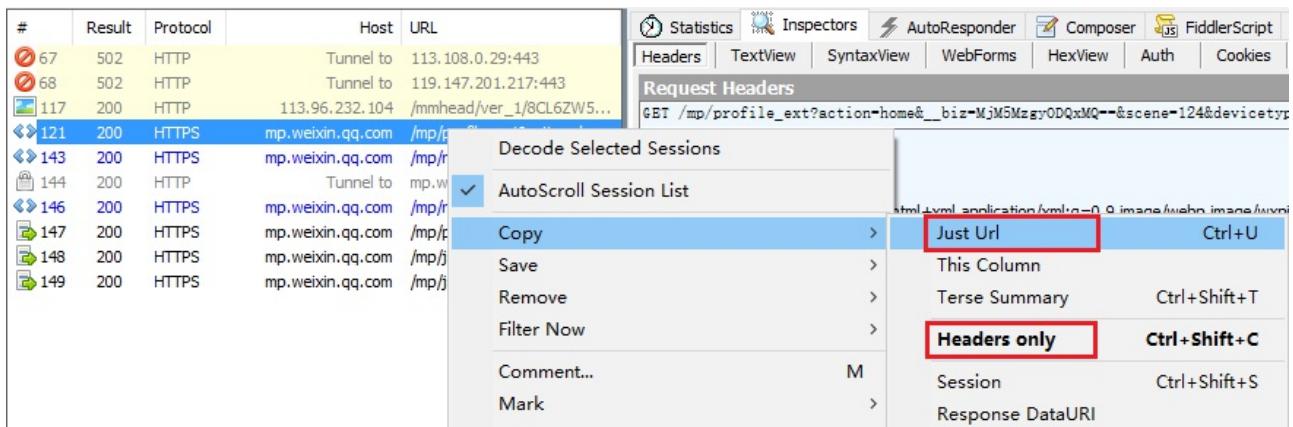


现在我们来编写一个最粗糙的版本，你需要做两件事：

- 1：找到完整URL请求地址
- 2：找到完整的请求头（headers）信息，Headers里面包括了

cookie、User-agent、Host 等信息。

我们直接从 Fiddler 请求中拷贝 URL 和 Headers，右键 -> Copy -> Just Url/Headers Only



最终拷贝出来的URL很长，它包含了很多的参数

```
url = "https://mp.weixin.qq.com/mp/profile_ext" \
    "?action=home" \
    "&__biz=MjM5MzgyODQxMQ==" \
    "&scene=124" \
    "&devicetype=android-24" \
    "&version=26051633&lang=zh_CN" \
    "&nettype=WIFI&a8scene=3" \
    \
"&pass_ticket=MXADI5SFjXvX7DFPRuUEJhWHEWvRha2x1Re
%2BoJkveUxIonMfnxY1kM9c0Pmm6JRx" \
    "&wx_header=1"
```

暂且不去分析（猜测）每个参数的意义，也不知道那些参数是必须的，总之我把这些参数全部提取出来。然后把 Headers 拷贝出来，发现 Fiddler 把 请求行、响应行、响应头都包括进来了，我们只需要中间的请求头部分。

```
GET https://mp.weixin.qq.com/mp/profile_ext?action=home&__biz=Mzg5ODQxMQ==8
Host: mp.weixin.qq.com
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 7.0; M1 E Build/NRD90M; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/60.0.3112.79 Mobile Safari/537.36
x-wechat-uin: NTI1NDc3NTE4
x-wechat-key: c37a3f1c3525d70e11e99aa435fc348e95c43d8c1bd700a92a64307c670c1479
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,en-US;q=0.8
Cookie: rewardsn=bfcf84efa5c0e2d0cc3b; wxtokenkey=bacede7644d9c17f50857845a1103
Q-UA2: QV=3&PL=ADR&PR=WX&PP=com.tencent.mm&PPVN=6.5.22&TBSVC=43602&CO=BK&COVC=0
Q-UUID: 0fd685fa8c515a30dd9f7caf13b788cb
Q-Auth: 31045b957cf33acf31e40be2f3e71c5217597676a9729f1b
```

请求行

请求头

```
HTTP/1.1 200 OK
```

响应行

```
Content-Type: text/html; charset=UTF-8
Cache-Control: no-cache, must-revalidate
Set-Cookie: wxuin=525477518; Path=/; HttpOnly
Set-Cookie: pass_ticket=MXADI5SFjXvX7DFPRuUEJhWHEWvRha2x1Re+oJkveUxIonMfnxY1kM9
```

响应头

因为 `requests.get` 方法里面的 `headers` 参数必须是字典对象，所以，先要写个函数把刚刚拷贝的字符串转换成字典对象。

```
def headers_to_dict(headers):  
    """  
    将字符串  
    ...  
    Host: mp.weixin.qq.com  
    Connection: keep-alive  
    Cache-Control: max-age=  
    ...  
    转换成字典对象  
    {  
        "Host": "mp.weixin.qq.com",  
        "Connection": "keep-alive",  
        "Cache-Control": "max-age"  
    }  
    :param headers: str  
    :return: dict  
    """  
  
    headers = headers.split("\n")  
    d_headers = dict()  
    for h in headers:  
        if h:  
            k, v = h.split(":", 1)  
            d_headers[k] = v.strip()  
    return d_headers
```

最终 v0.1 版本出来了，不出意外的话，公众号历史文章数据就在 `response.text` 中。如果返回的内容非常短，而且 `title` 标签是 `<title>验证</title>`，那么说明你的请求参数或者请求头有误，最有可能的一种请求就是 `Headers` 里面的 `Cookie` 字段过期，从手机微信端重新发起一次请求获取最新的请求参数和请求头试试。

```
# v0.1
def crawl():
    url = "https://mp.weixin.qq.com/..." # 省略了
    headers = "" # 省略了
Host: mp.weixin.qq.com
Connection: keep-alive
Upgrade-Insecure-Requests: 1
"""
headers = headers_to_dict(headers)
response = requests.get(url, headers=headers,
verify=False)
print(response.text)
```

最后，我们顺带把响应结果另存为html文件，以便后面重复使用，分析里面的内容

```
with open("weixin_history.html", "w",
encoding="utf-8") as f:
    f.write(response.text)
```

用浏览器打开 weixin_history.html 文件，查看该页面的源代码，搜索微信历史文章标题的关键字 "11月赠书"（就是我以往发的文章），你会发现，历史文章封装在叫 msgList 的数组中（实际上该数组包装在字典结构中），这是一个 Json 格式的数据，但是里面还有 html 转义字符需要处理

```
348 var headingimg = "http://wx.qlogo.cn/mmhead/Q3auHgzwzM6t
349 || "";
350 var nickname = "Python之禅" || "";
351 var is_banned = "0" * 1;
352 var __biz = "MjM5MzgyODQxMQ==";
353 var next_offset = "10" * 1;
354 var use_demo = "0" * 1;
355
356 var msgList = ["list": [{"comm_msg_info": {
357 "id": "1000000164", "type": 49, "datetime": "1512177599", "fakeid": "2393828411", "status": 2, "content": "", "app_msg_ext_info": {
358 "title": "11月赠书福利中奖结果出炉", "digest": "如图", "content": "", "fileid": "502883915", "content_url": "http://mp.weixin.qq.com/s?__biz=MjM5MzgyODQxMQ==&mid=2650367566&idx=1&sr=692295d1c53bd4054adfc34c71b1073&chksm=be9ccdd1a89eb540c383985ed857b6e56394e44d3f6dd304bbc1f77647f050de911f608904f88&sc=27#wechat_redirect", "source_url": "", "cover": "http://\\V\\Vmbi.pic.cn\\mmbiz.jpg\\r01libUkmNGMnjCSYuBszzITUp9JG9CicYW8qt3qKf7gcZDayBScJzTBarJP2z1Uupq6dZ68qHR82VzK-2201"}, "content": ""}], "content": ""}], "content": ""}]
```

接下来我们就来写一个方法提取出历史文章数据，分三个步骤，首先用正则提取数据内容，然后 html 转义处理，最终得到一个列表对象，返回最近发布的10篇文章。

```
def extract_data(html_content):  
    """  
    从html页面中提取历史文章数据  
    :param html_content 页面源代码  
    :return: 历史文章列表  
    """  
  
    import re  
    import html  
    import json  
  
    rex = "msgList = '(.*)'"  
    pattern = re.compile(pattern=rex, flags=re.S)  
    match = pattern.search(html_content)  
    if match:  
        data = match.group(1)  
        data = html.unescape(data)  
        data = json.loads(data)  
        articles = data.get("list")  
        for item in articles:  
            print(item)  
    return articles
```

最终提取出来的数据总共有10条，就是最近发表的10条数据，我们看看每条数据返回有哪些字段。

```
article = {'app_msg_ext_info':  
          {'title': '11月赠书，总共10本，附  
          Python书单',  
           'copyright_stat': 11,  
           'is_multi': 1,  
           'content': '',  
           'author': '刘志军',  
           'subtype': 9,}
```

```
        'del_flag': 1,
        'fileid': 502883895,
        'content_url':
'http://\mp.weixin.qq.com...',  

        ''  

        'digest': '十一月份赠书福利如期而至,  

更多惊喜等着你',  

        'cover':  

'http://\mmbiz.qpic.cn\...',  

        'multi_app_msg_item_list':  

[{'fileid': 861719336,  

'content_url': 'http://\mp.weixin.qq.com',  

'content': '', 'copyright_stat': 11,  

'cover': 'http://\mmbiz.qpic.cn',  

'del_flag': 1,  

'digest': '多数情况下，人是种短视的动物',  

'source_url': '',  

'title': '罗胖60秒：诺贝尔奖设立时，为何会被骂？',  

'author': '罗振宇'  

    },  

        'source_url':  

'https://github.com\'  

    },  

        'comm_msg_info': {'datetime': 1511827200,  

        'status': 2,
```

```
'id': 1000000161,  
'fakeid': '2393828411',  
'content': '',  
'type': 49}}}
```

我们结合下面这张图来看：

星期三 22:08



今年，谁又“暴富”了？ | 2017内容创业
融资盘点

经历5年才“转正”的公众号，
2018年还有哪些可能性？



学霸时尚博主怼北京，北京真不适合搞时尚？ | 新榜11月网红排...



这三堂课，性价比高到为商家担心啊！ | 新榜有赚

新榜
有赚

微信“搜一搜”可以查地图了，还能领红包；7名陌陌主播开赛，...



上面这张图是微信作为单次推送发给用户的多图文消息，有发送时间对应comm_msg_info.datetime, app_msg_ext_info中的字段信息就是第一篇文章的字段信息，分别对应：

- title: 文章标题

- content_url: 文章链接
- source_url: 原文链接, 有可能为空
- digest: 摘要
- cover: 封面图
- datetime: 推送时间

后面几篇文章以列表的形式保存在 `multi_app_msg_item_list` 字段中。

到此, 公众号文章的基本信息就抓到了, 但也仅仅只是公众号的前 10 条推送, 如何获取更多历史文章等问题放在下节讲解。

本节完整代码可以在 GitHub 仓库 [weixincrawler_v
\(https://github.com/pythonzhichan/weixincrawler/tree/v0.1\)](https://github.com/pythonzhichan/weixincrawler/tree/v0.1) 查看。