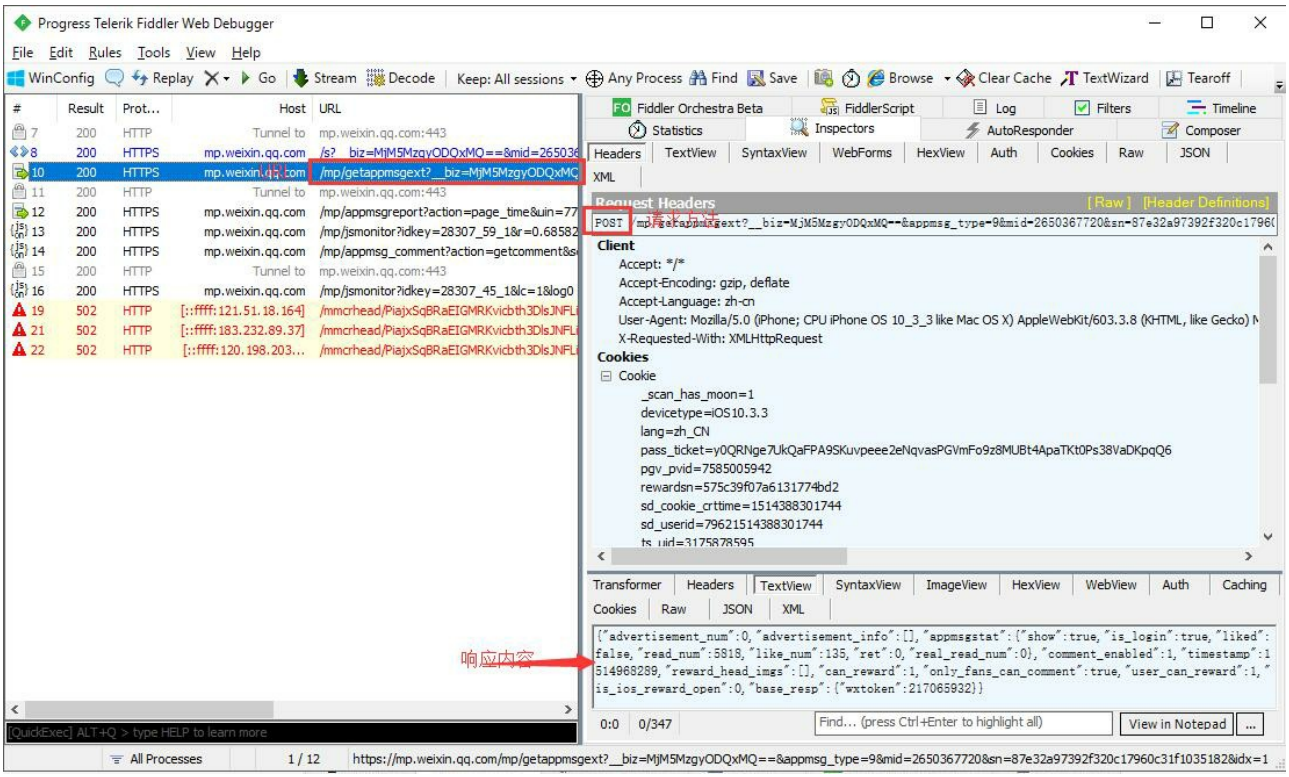
获取文章阅读数、点赞数、评论数、赞赏数

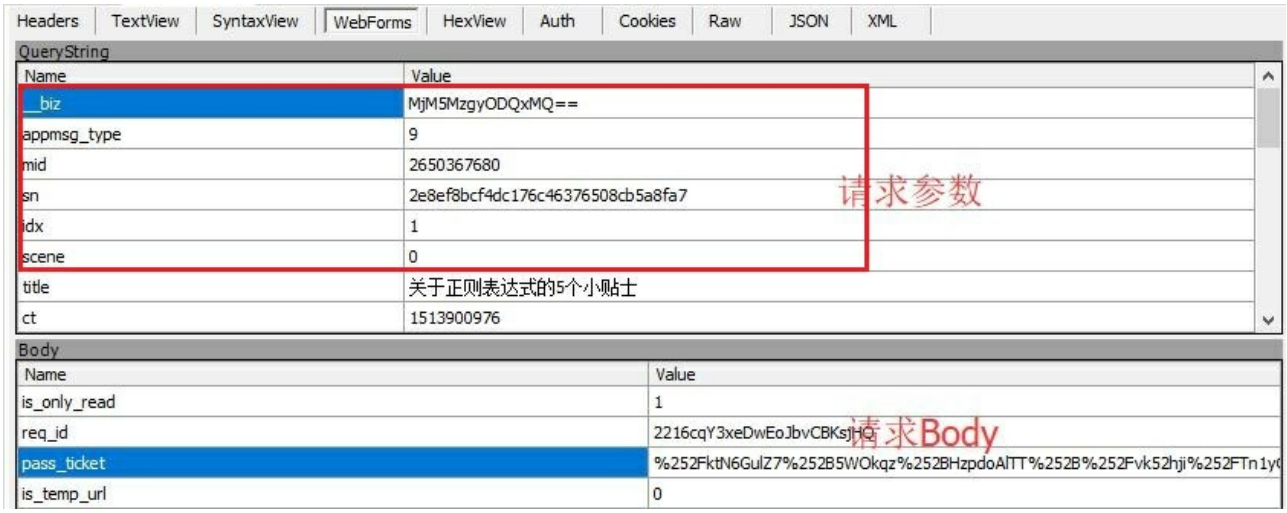
如果只是获取所有文章的基本信息价值并不大，最多能对文章做检索，只有得到文章的阅读数、点赞数、评论数和赞赏数之后数据才有数据分析的价值。这节就来讨论如何获取这些数据。

抓包分析

点开任意一篇文章，通过 Fiddler 或 Charles 抓包分析，逐个分析每个请求，通过观察发现获取文章阅读数、点赞数的URL接口为（我们命名为 data_url）：https://mp.weixin.qq.com/mp/getappmsgext ，后面有很多查询参数，请求方法为 POST



该请求的查询参数有28个之多，另外还附有请求 Body。



返回的响应数据是JSON格式，根据字段名称基本能猜出其中的意义，阅读数、点赞数、赞赏数都包含在其中

```
{
    "advertisement_num": 0,
    "advertisement_info": [ ],
    "appmsgstat": {
        "show": true,
        "is_login": true,
        "liked": false,
        "read_num": 6395,   # 阅读数
        "like_num": 190,    # 点赞数
        "ret": 0,
        "real_read_num": 0
    },
    "comment_enabled": 1,
    "timestamp": 1514972862,
    "reward_head_imgs": [  # 赞赏头像列表

"http://wx.qlogo.cn/mmhead/V3bYdzb7P4DLf3e7Xf74qSicESO8QdeupE5ibs8YI6xibE/132",

"http://wx.qlogo.cn/mmhead/Q3auHgzwzM7KF8PIsOicjLuRpsRzFhibeKs3sHFJGKkxDguAnF2gQJdA/132",
```

```
    ],
    "reward_total_count": 16,  # 赞赏数
    "can_reward": 1,
    "only_fans_can_comment": true,
    "user_can_reward": 1,
    "reward_qrcode_ticket":
"%2B%2FfLw%2BXXGQwDD0ik6GwpMhSzLBMFCkwhjpXhStXNjXo%3D",
    "base_resp": {
        "wxtoken": 723698581
    }
}
```

确定了请求的URL及查询参数，请求方法，请求体，请求头也能查看到，返回的数据也有了，剩下的问题是如何批量获取不同文章的数据，这需要从请求的 data_url 着手分析。

为了找出 data_url 中查询参数的规律，先对比文章详情的 content_url （就是在上一节得到的文章详情URL）

```
# 文章的URL
content_url = "http://mp.weixin.qq.com/s?" \
              "__biz=MjM5MzgyODQxMQ==&" \
              "mid=2650367413&idx=1&" \

"sn=637de06b162c21605eef3db41ee4a1bb&" \

"chksm=be9cdee189eb57f78994371ce1b5b42656bf77160eeee592507df06bae0cea58b542aeabe0a4&" \
              "scene=27"
```

不得而知，__biz，mid，idx，sn，scene，chksm 是构成一篇文章的完整URL，而文章阅读数的URL是：

```
# 阅读数URL
data_url =
"https://mp.weixin.qq.com/mp/getappmsgext?" \
            "__biz=MjM5MzgyODQxMQ==&" \
            "appmsg_type=9&" \
            "mid=2650367720&" \
            "sn=87e32a97392f320c17960c31f1035182&"
\
            "idx=1&" \
            "scene=27&" \

"title=2018%20%E5%B9%B4%EF%BC%8C%E5%AD%A6%E7%82%B
9%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD%EF%BC%88%E8
%B5%A0%E4%B9%A68%E6%9C%AC%EF%BC%89&" \
            "ct=1514505600&" \

"abtest_cookie=AwABAAoADAANAAkAJIgeAGKIHgD8iB4Ab4
keAPiJHgAHih4AD4oeAEyKHgBdih4AAAA=&" \
            "devicetype=iOS10.3.3&" \

"version=/mmbizwap/zh_CN/htmledition/js/appmsg/in
dex3a9713.js&" \
            "f=json&" \
            "r=0.34167962392788 9&is_need_ad=0&" \
            "comment_id=2810810222&" \
            "is_need_reward=1&" \
            "both_ad=0&" \
            "reward_uin_count=27&" \
            "msg_daily_idx=1&" \
            "is_original=0&" \
            "uin=777&" \
            "key=777&" \
```

```
"pass_ticket=y0QRNge7UkQaFPA9SKuvpeee2eNqvasPGVmFo9z8MUBt4ApaTKt0Ps38VaDKpqQ6&" \
            "wxtoken=2170659328&" \
            "devicetype=iOS10.3.3&" \
            "clientversion=16060123&" \

"appmsg_token=938_vSjt2FisNybtpLZCYw5DGyn0L2PK7qpCkzVlZySGKUjQDC0UMw3SNoS1Atum66a7ElZYWWb5amRtAy8m&" \
            "x5=0&" \
            "f=json"
```

对比两个URL，你会发现 content_url 中的参数除了 chksm 其它几个参数都在 data_url 中，我们把 content_url 中的参数替换到 data_url 再来验证请求会不会正常返回数据。至于其他参数要不要改，怎么改我们先放一边（这是一个不断猜想、验证的过程，经过我的多次试验，除了 appmsg_token 有一定的时效之外，其它值可以保持不变，也就是说不同的文章，只要把content_url中的参数替换到 data_url 中就可以获取该文章的数据了。）

代码实现

```
    @staticmethod
    def update_post(post):
        """
        post 参数是从mongodb读取出来的一条数据
        稍后就是对这个对象进行更新保存
        :param post:
        :return:
        """

        # 这个参数是我从Fiddler中拷贝出 URL，然后提取出查询参数部分再转换成字典对象
```

```
        # 稍后会作为参数传给request.post方法
        data_url_params = {'__biz':
'MjM5MzgyODQxMQ==', 'appmsg_type': '9', 'mid':
'2650367727',
                           'sn':
'08ce54f6f36873e74c638421012bb495', 'idx': '1',
'scene': '0',
                           'title':
'2017%E5%B9%B4%EF%BC%8C%E6%84%9F%E8%B0%A2%E4%BD%A
0%E4%BB%AC%EF%BC%8C2018%E5%B9%B4%EF%BC%8C%E6%88%9
1%E4%BB%AC%E7%BB%A7%E7%BB%AD%E5%8A%AA%E5%8A%9B%E5
%89%8D%E8%A1%8C',
                           'ct': '1514796292',
                           'abtest_cookie':
'AwABAAoADAANAAgAJIgeALuIHgDhiB4A/IgeAPqJHgANih4A
TYoeAF6KHgAAAA==',
                           'devicetype':
'android-24',
                           'version':
'/mmbizwap/zh_CN/htmledition/js/appmsg/index3a971
3.js', 'f': 'json',
                           'r':
'0.6452677228890584', 'is_need_ad': '1',
'comment_id': '1741225191',
                           'is_need_reward': '1',
'both_ad': '0', 'reward_uin_count': '24',
'msg_daily_idx': '1',
                           'is_original': '0',
'uin': '777', 'key': '777',
                           'pass_ticket':
'mXHYjLnkYux1rXx8BxNrZpgW4W%252ByLZxcuvpDWlxbBrjv
Jo3ECB%252BckDAsy%252FTJJK6P',
                           'wxtoken':
```

```
'1805512665', 'clientversion': '26060133',
                                'appmsg_token':
'938_VN3Rr7O4RIU7lm%2F8_amSJbZBo3RJXACjIMDwDu5ZPb
Sm2_SW6RpnZGb2Vrp6ECxr9y5QoVCI7H-iQotJ',
                                'x5': '1'}

        # url转义处理
        content_url =
html.unescape(post.content_url)
        # 截取content_url的查询参数部分
        content_url_params =
urlsplit(content_url).query
        # 将参数转化为字典类型
        content_url_params =
utils.str_to_dict(content_url_params, "&", "=")
        # 更新到data_url

data_url_params.update(content_url_params)
        body =
"is_only_read=1&req_id=03230SZyTR8kQlPVkKwxbt1A&p
ass_ticket=mXHYjLnkYux1rXx8BxNrZpgW4W%25252ByLZxc
uvpDWlxbBrjvJo3ECB%25252BckDAsy%25252FTJJK6P&is_t
emp_url=0"
        data = utils.str_to_dict(body, "&", "=")

        headers = """
Host: mp.weixin.qq.com
Connection: keep-alive
Content-Length: 155
Origin: https://mp.weixin.qq.com
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Linux; Android 7.0; M1 E
Build/NRD90M; wv) AppleWebKit/537.36 (KHTML, like
```

```
Gecko) Version/4.0 Chrome/53.0.2785.49 Mobile
MQQBrowser/6.2 TBS/043632 Safari/537.36
MicroMessenger/6.6.1.1220(0x26060133)
NetType/WIFI Language/zh_CN
Content-Type: application/x-www-form-urlencoded;
charset=UTF-8
Accept: */*
Referer: https://mp.weixin.qq.com/s?
__biz=MjM5MzgyODQxMQ==&mid=2650367727&idx=1&sn=08
ce54f6f36873e74c638421012bb495&chksm=be9cddbb89eb
54ad436af5c27c0d0db06da7e3aec613a33dd99f935d684a7
7b555241207f1ba&scene=0&ascene=7&devicetype=andro
id-
24&version=26060133&nettype=WIFI&abtest_cookie=Aw
ABAAoADAANAAgAJIgeALuIHgDhiB4A%2FIgeAPqJHgANih4AT
YoeAF6KHgAAAA%3D%3D&lang=zh_CN&pass_ticket=mXHYjL
nkYux1rXx8BxNrZpgW4W%2ByLZxcuvpDWlxbBrjvJo3ECB%2B
ckDAsy%2FTJJK6P&wx_header=1
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,en-US;q=0.8
Cookie: rewardsn=05c38771473771b68376;
wxtokenkey=92c034f1d4d5cfe011a9222522d96c3af508a6
e35160b5f6fefa185431bda832; wxuin=525477518;
devicetype=android-24; version=26060133;
lang=zh_CN;
pass_ticket=mXHYjLnkYux1rXx8BxNrZpgW4W+yLZxcuvpDW
lxbBrjvJo3ECB+ckDAsy/TJJK6P;
wap_sid2=CI7NyPoBElx2ZFNJVXFOVFh2S3U5X1hLS2pZb2Z0
Ujd1NTBPdlMzbEpwMjdVRlYtTHluRWkwZzIwUzY4ZVM3Y294M
zU5aDM5eWxfRWVKOVJoY0dvVmZuQTk2S1JLS29EQUFBfjCQ5L
PSBTgNQAE=
Q-UA2:
QV=3&PL=ADR&PR=WX&PP=com.tencent.mm&PPVN=6.6.1&TB
```

```
SVC=43602&CO=BK&COVC=043632&PB=GE&VE=GA&DE=PHONE&
CHID=0&LCID=9422&MO= M1E
&RL=1080*1920&OS=7.0&API=24
Q-GUID: 0fd685fa8c515a30dd9f7caf13b788cb
Q-Auth:
31045b957cf33acf31e40be2f3e71c5217597676a9729f1b
        """

        headers = utils.str_to_dict(headers)

        data_url =
"https://mp.weixin.qq.com/mp/getappmsgext"

        r = requests.post(data_url, data=data,
verify=False, params=data_url_params,
headers=headers)

        result = r.json()
        if result.get("appmsgstat"):
            post['read_num'] =
result.get("appmsgstat").get("read_num")
            post['like_num'] =
result.get("appmsgstat").get("like_num")
            post['reward_num'] =
result.get("reward_total_count")
            post['u_date'] = datetime.now()
            logger.info("「%s」read_num: %s
like_num: %s reward_num: %s" %
                        (post.title,
post['read_num'], post['like_num'],
post['reward_num']))
            post.save()
        else:
```

```
                logger.warning(u"没有获取的真实数据，请检查请求参数是否正确，返回的数据为：data=%s" % r.text)
```

需要注意的是 iOS 没有赞赏功能，所以如果要获取赞赏数据，我们必须用 Android 设备来抓取数据。现在就来遍历更新每条数据的内容：

```
  crawler = WeiXinCrawler()
    for post in Post.objects(read_num=0):
        crawler.update_post(post)
        time.sleep(1) # 防止恶意刷
```

不出意外的话，能正常获取到数据，在抓取的过程中，微信会有反爬虫限制，爬了一段时间后，返回的数据成了：

```
{"base_resp":{"ret":301,"errmsg":"default"}}
```

这个时候需要休息一会儿才能继续爬虫，换IP也没用，因为微信会根据你的微信账号进行限制，如果需要大规模爬虫就有必要准备多个微信号来操作。本节完整代码：[weixincrawler-v0.4 (https://github.com/pythonzhichan/weixincrawler/tree/v0.4)](https://github.com/pythonzhichan/weixincrawler/tree/v0.4)