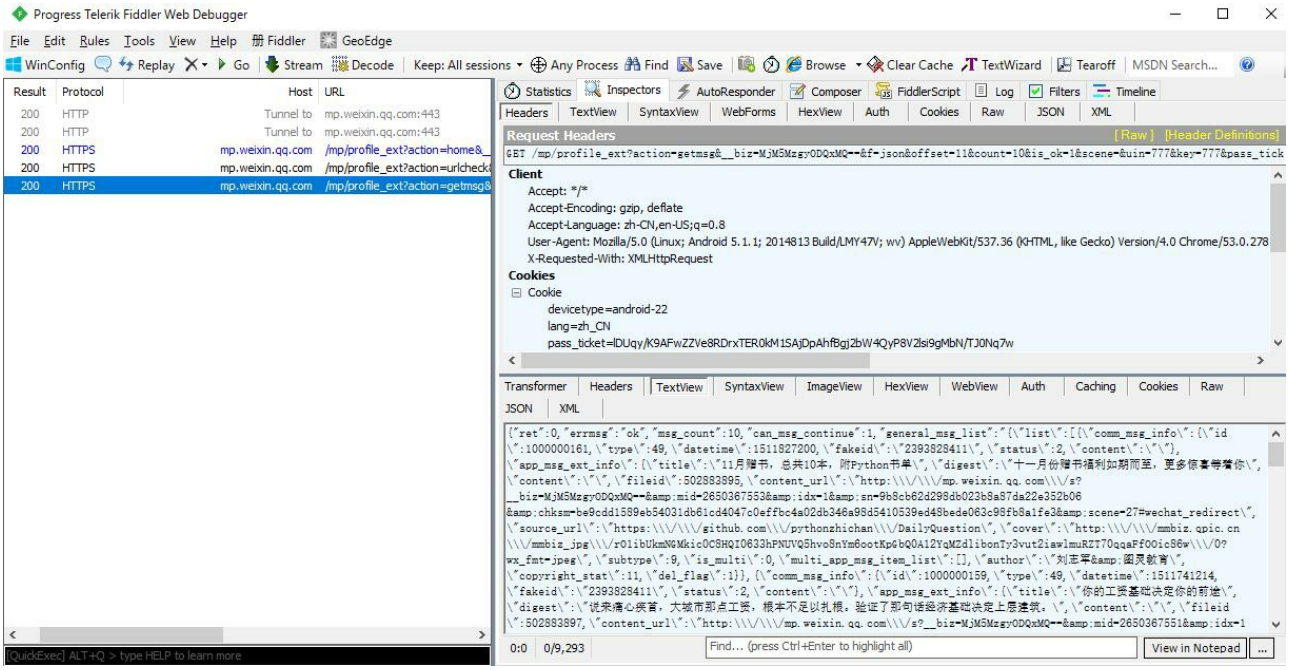
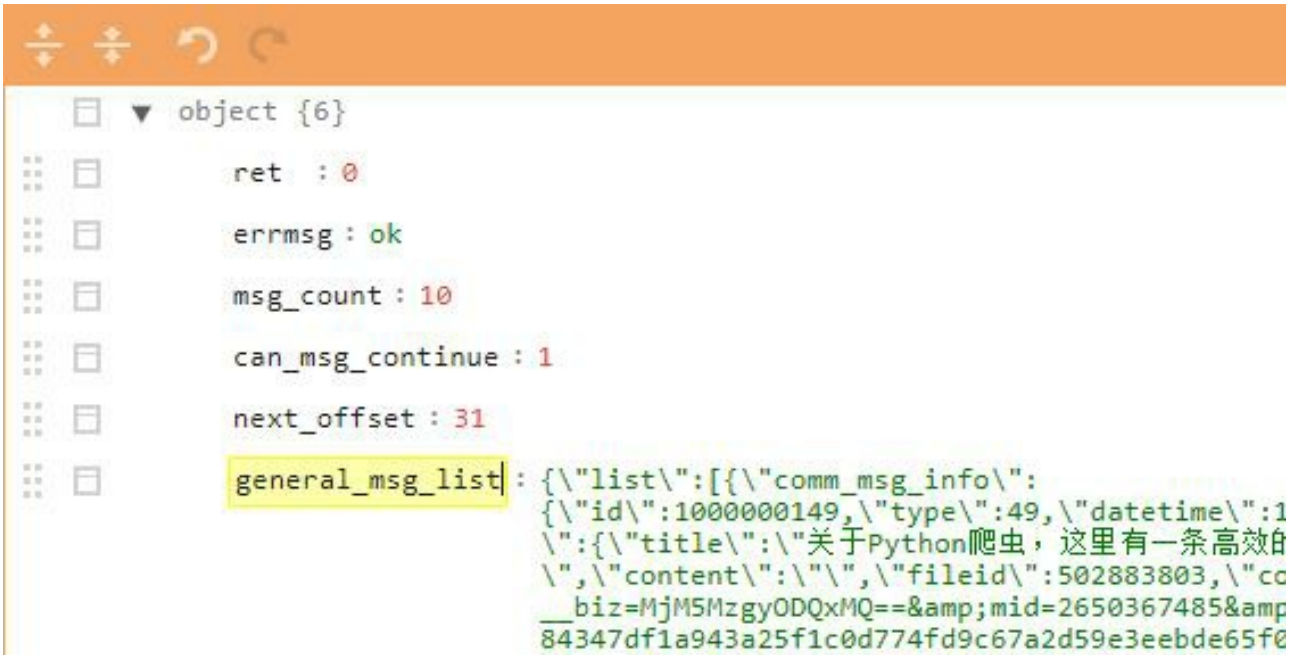
抓取公众号所有历史文章

我们按照第三节使用 Fiddler 抓包方式，打开手机某个微信公众号历史文章列表，上拉加载更多，找到加载更多文章的 URL 请求地址，你会看到 Fiddler 会有一个加载更多文章列表的请求。



分析抓包数据

该接口返回的数据是 JSON 格式，这种数据格式处理起来非常方便，首先我们把数据拷贝到 Chrome 插件 JSON Editor 或者找一个[JSON Online Formatter (https://jsonformatter.curiousconcept.com/)](https://jsonformatter.curiousconcept.com/) 对返回的数据进行格式化处理，以便查看每个字段所代表的意义。



你可以大概猜出来每个字段的意思

- ret：请求是否成功，0就表示成功
- msg_count： 返回的数据条数
- can_msg_continue： 是否还有下一页数据
- next_offset： 下一次请求的起始位置
- general_msg_list：真实数据

general_msg_list是历史文章里面的基本信息，包括每篇文章的标题、发布时间、摘要、链接地址、封面图等，而像文章的阅读数、点赞数、评论数、赞赏数这些数据都需要通过额外接口获取。

代码实现

分析完后，用代码实现其实非常简单，按照上节的方式，我们把 URL 和 Header 信息直接从 Fiddler 中拷贝过来。

```
# crawler.py
# -*- coding: utf-8 -*-

import logging
import utils
```

```
import requests

logging.basicConfig(level=logging.INFO)

logger = logging.getLogger(__name__)


class WeiXinCrawler:
    def crawl(self):
        """
        爬取更多文章
        :return:
        """
        url =
"https://mp.weixin.qq.com/mp/profile_ext?" \
                "action=getmsg&" \
                "__biz=MjM5MzgyODQxMQ==" \
                "&f=json&" \
                "offset=11&" \
                "count=10&" \
                "is_ok=1" \
                "&scene=124&" \
                "uin=777&key=777&" \

"pass_ticket=25llsA6zWUPC9KHOvP4oE%2BQwJ3nS%2F3Cj
eWxeKBjDhxCb7V1lQQJa6d0ZrgSmCvWa&wxtoken=&" \

"appmsg_token=936_qKN8I1KSEO%252BWB2YUShHV8kgkIGX
gzl-CT8JJpw~~&" \
                "x5=0&" \
                "f=json"
```

```
        headers = """
                    Host: mp.weixin.qq.com
                    .... 省略了，自己补充 ...
                    """
        headers = utils.str_to_dict(headers)
        response = requests.get(url,
headers=headers, verify=False)
        result = response.json()
        if result.get("ret") == 0:
            msg_list =
result.get("general_msg_list")
            logger.info("抓取数据: offset=%s,
data=%s" % (offset, msg_list))
        else:
            # 错误消息
            # {"ret":-3,"errmsg":"no
session","cookie_count":1}
            logger.error("无法正确获取内容，请重新从
Fiddler获取请求参数和请求头")
            exit()


if __name__ == '__main__':
    crawler = WeiXinCrawler()
    crawler.crawl()
```

成功爬取了第二页的数据，那么第三页呢，第四页呢？所以，我们还需要对该方法进行重构，使得它可以抓取公众号全部历史文章。通过字段 can_msg_continue 确定是否继续抓取，再结合 next_offset 就可以加载更多数据，我们需要把 url 中可变的参数 offset 用变量来代替，递归调用直到 can_msg_continue 为 0 说明所有文章都爬取完了。

```
    def crawl(self, offset=0):
        """
        爬取更多文章
        :return:
        """
        url =
"https://mp.weixin.qq.com/mp/profile_ext?" \
              "action=getmsg&" \
              "__biz=MjM5MzgyODQxMQ==&" \
              "f=json&" \
              "offset={offset}&" \
              "count=10&" \
              "is_ok=1&" \
              "scene=&" \
              "uin=777&" \
              "key=777&" \

"pass_ticket=25llsA6zWUPC9KHOvP4oE+QwJ3nS/3CjeWxe
KBjDhxCb7V1lQQJa6d0ZrgSmCvWa&" \
              "wxtoken=&" \

"appmsg_token=936_qKN8I1KSEO%2BWB2YUShHV8kgkIGXgz
l-CT8JJpw~~&" \
              "x5=1&" \
              "f=json".format(offset=offset)  # 请
将appmsg_token和pass_ticket替换成你自己的

        headers = """
Host: mp.weixin.qq.com
.... 省略了，自己补充 ...
"""
        headers = utils.str_to_dict(headers)
        response = requests.get(url,
```

```
headers=headers, verify=False)
            result = response.json()
            if result.get("ret") == 0:
                msg_list =
result.get("general_msg_list")
                logger.info("抓取数据：offset=%s,
data=%s" % (offset, msg_list))
                # 递归调用
                has_next =
result.get("can_msg_continue")
                if has_next == 1:
                    next_offset =
result.get("next_offset")
                    time.sleep(2)
                    self.crawl(next_offset)
            else:
                # 错误消息
                # {"ret":-3,"errmsg":"no
session","cookie_count":1}
                logger.error("无法正确获取内容，请重新从
Fiddler获取请求参数和请求头")
                exit()
```

当 has_next 为 0 时，说明已经到了最后一页，这时才算爬完了一个公众号的所有历史文章，现在把所有的文章摘要数据抓取下来了，但是数据还没有存储，下一节，我们将使用MongoDB将数据进行持久化。

本节完整代码地址：[weixincrawler/v0.2 (https://github.com/pythonzhichan/weixincrawler/tree/v0.2)](https://github.com/pythonzhichan/weixincrawler/tree/v0.2)