# 主要思路

帖子：

https://www.jianshu.com/p/8dc04794e35f

登录微博

--》我的微博主页

https://m.weibo.cn/u/3628313085

--》我的微博账号ID

3628313085

访问微博主页

--》访问某个账号的微博主页

https://m.weibo.cn/u/{对方微博ID}

--》以GitHubDaily为例 ID为5722964389

https://m.weibo.cn/u/5722964389



访问微博

--》访问该用户的某一条微博



--》跳转至

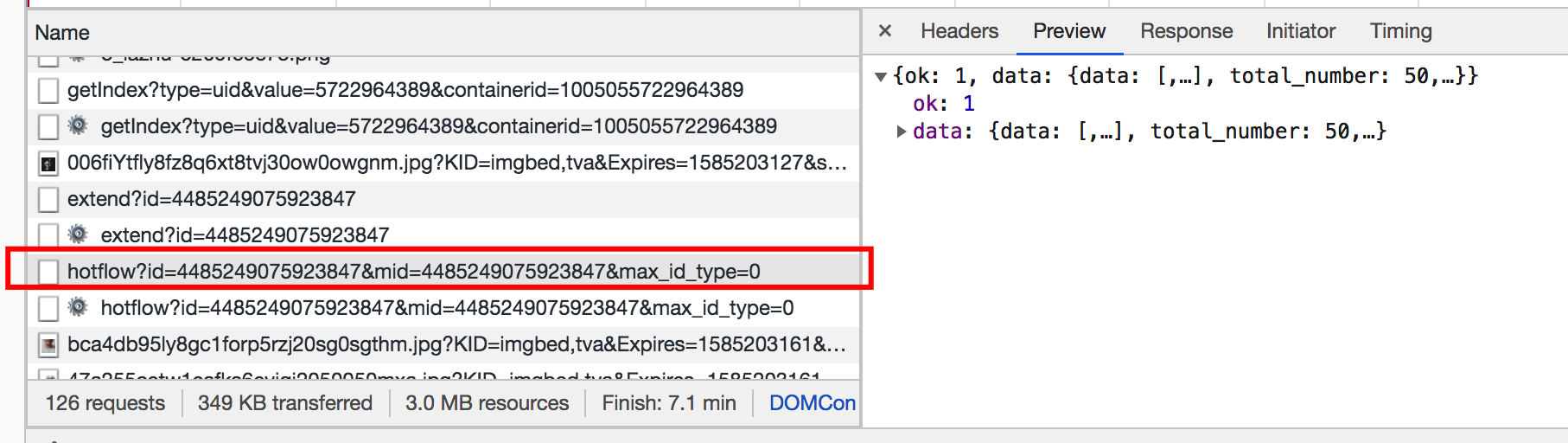
https://m.weibo.cn/detail/4485249075923847

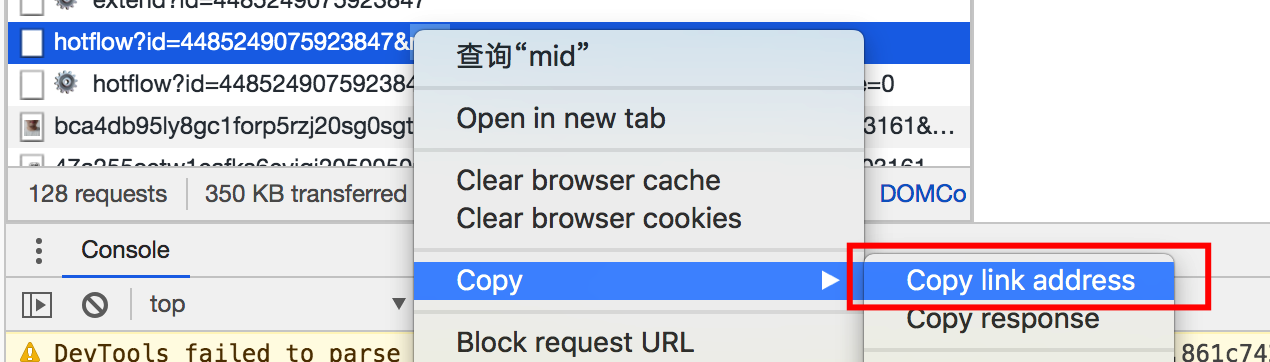
--》格式为

https://m.weibo.cn/detail/{微博ID}

分析微博评论

--》获得该条微博的评论URL





https://m.weibo.cn/comments/hotflow?id=4485249075923847&mid=4485249075923847&max\_id\_type=0

--》分析该条微博的评论URL

https://m.weibo.cn/comments/hotflow?id=4485249075923847&mid=4485249075923847&max\_id\_type=0

是该条微博的第一页评论

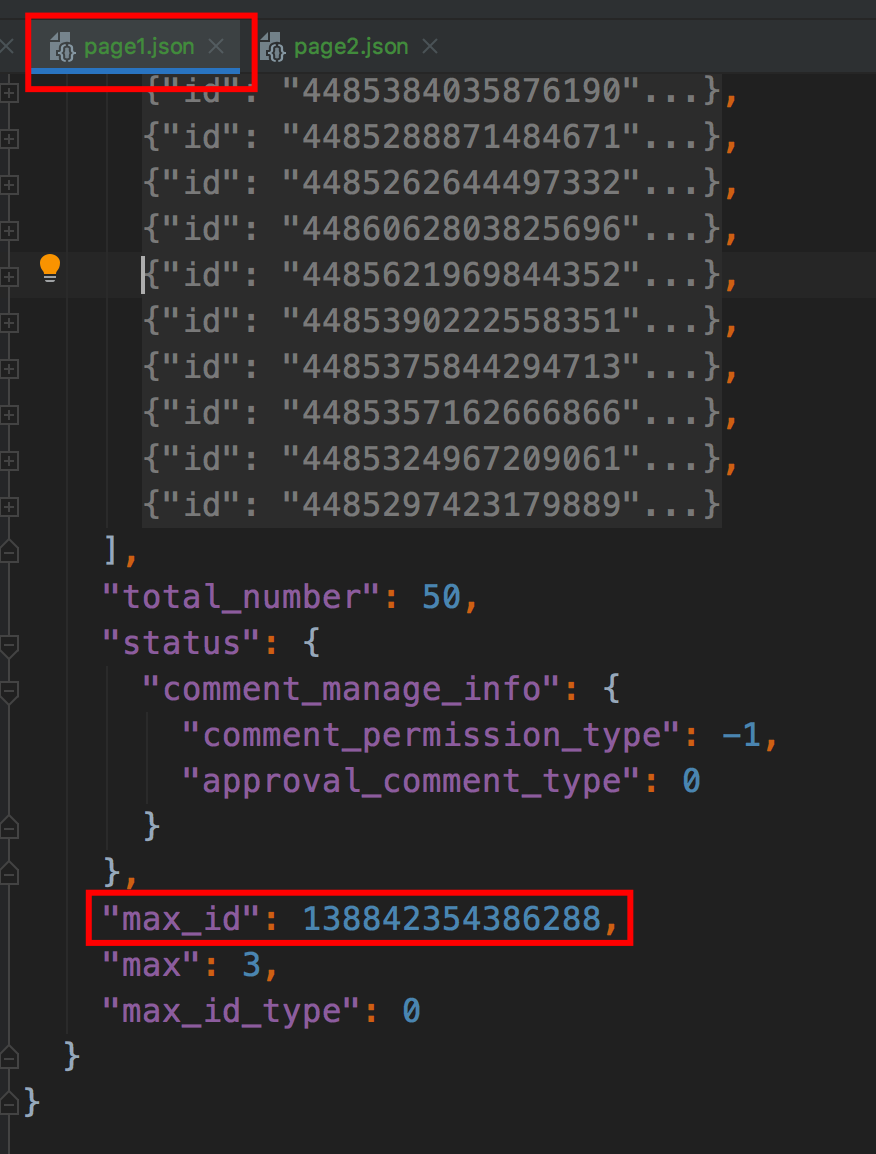
其中，各参数为：

id，mid，均为该条微博的id，4485249075923847

max\_id\_type，

访问第一页评论的response为：

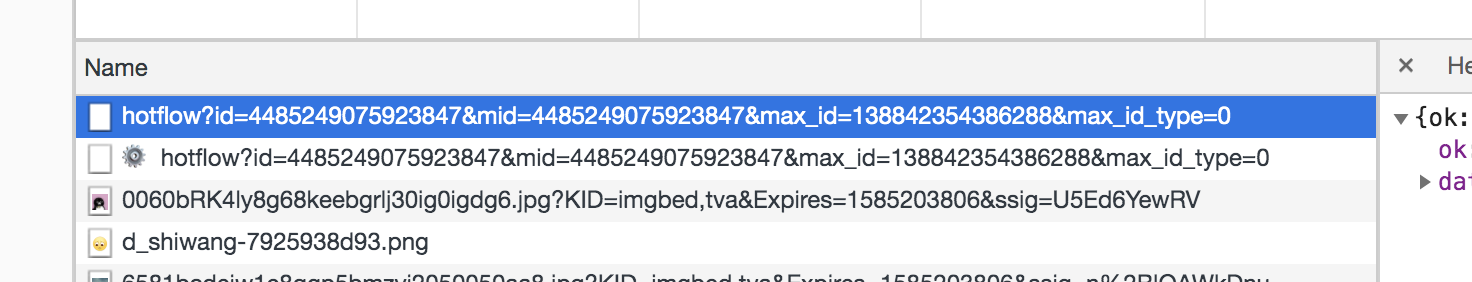
见page1.json



第一页返回的response里含有max\_id字段138842354386288

当我们继续往下翻评论，当翻到第二页评论时，发现URL有所变化

https://m.weibo.cn/comments/hotflow?id=4485249075923847&mid=4485249075923847&max\_id=138842354386288&max\_id\_type=0



其中，各参数为：

id，mid，均为该条微博的id，4485249075923847

max\_id，正是第一页返回的response中的max\_id字段

max\_id\_type，

访问第二页评论的response为：

见page2.json

总结

（1）在response中的json数据中有max\_id字段；

（2）评论第一页的url中没有max\_id字段

（3）评论第一页的url是可以反复多次打开的。

（4）评论第一页的response中的max\_id即是加载第二页评论的url

除了评论数据的第一页的url上不带有max\_id字段，其他的请求发起时候时，均需携带有上次请求response中返回的max\_id数据。例如第二次请求需要携带第一次请求返回的max\_id，第三次请求需要携带第二次请求返回的max\_id……，依次下去。同时这个顺序是不可逆的，比如当前已经返回了第六页的数据了，此时我们再想访问第三页的数据是不允许的（因为正常操作也是不会出现这种情况滴），而且对同一页的数据进行快速连续访问多次也是不合理的，后台小哥一样会丢给你一个{‘ok’:0}

**代码思路**

首先要爬评论是需要先登陆的，通过登陆获取cookie，之后的请求均携带此cookie。有了cookie了，接下来就是按照上面分析的规则进行评论的爬爬爬爬爬爬爬爬爬，先简单的用Request来走一波，改天上个scrapy版本/scrapy-redis版本。

作者：菇凉爱学习

链接：https://www.jianshu.com/p/8dc04794e35f

来源：简书

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。