

jmeter 接口压力测试

作者: [branda2019wj](#)

原文链接: <https://ld246.com/article/1555638677959>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

前言

笔者前面的blog主要介绍了jmeter如何请求接口，重点介绍了接口测试的方法之一post请求。今天笔者准备简单介绍下用jmeter做简单接口压力测试

笔者之前在本地搭建了慧测环境，所以就拿discuz论坛的登录接口举例说明

第一步：通过抓包获得discuz论坛的登录接口

笔者此处用的抓包工具是Fiddler，通过抓包，显示是post接口，地址如下：

`http://192.168.0.218/discuz/upload/member.php?mod=logging&action=login&loginsubmit=yes&infloat=yes&lssubmit=yes&inajax=1`

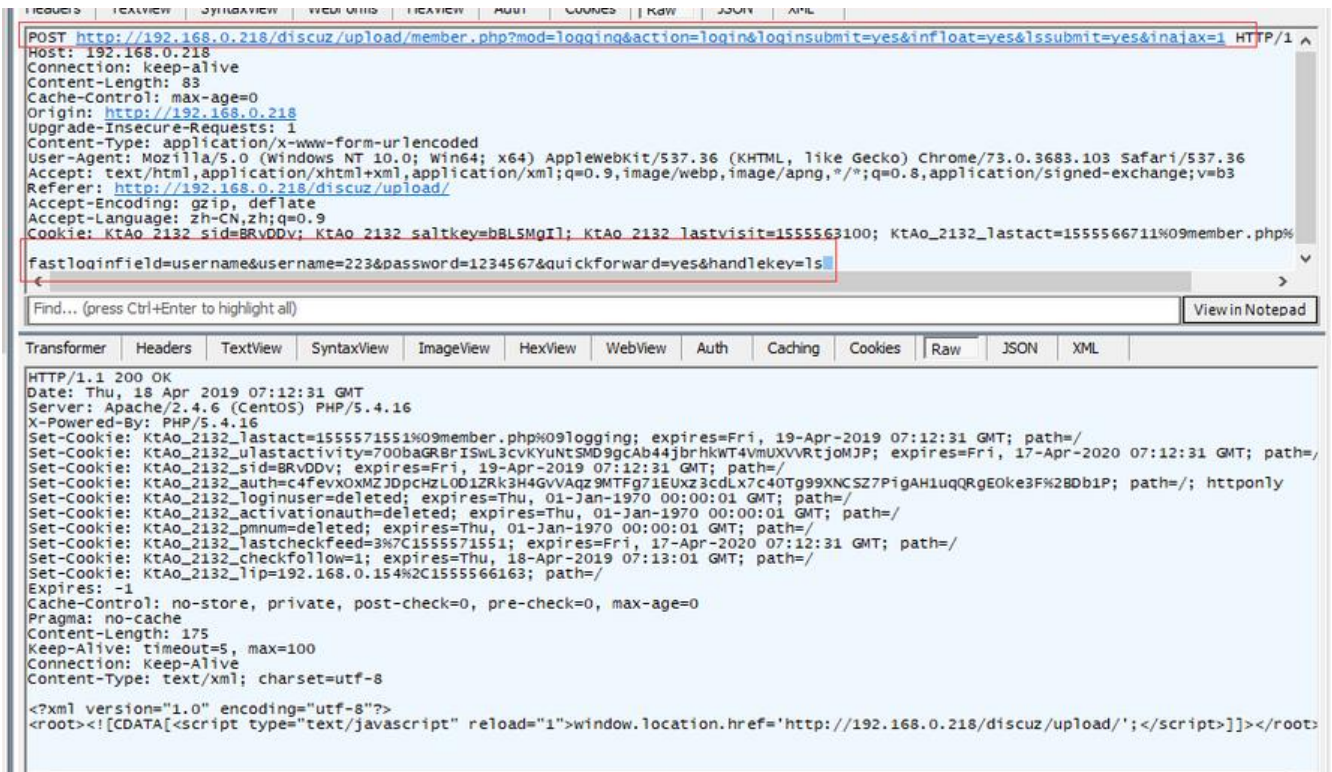
我们在下图的WebForms中可以看到QueryString和Body，这里的QueryString就是上面接口地址中后面跟的参数，Body指的是请求体，里面包括登录接口的账号和密码，但我们在上面的接口地址中没有看到Body中的内容，但在下方可以看到，或者我们直接到WebForms中查看就行

所有参数如下：

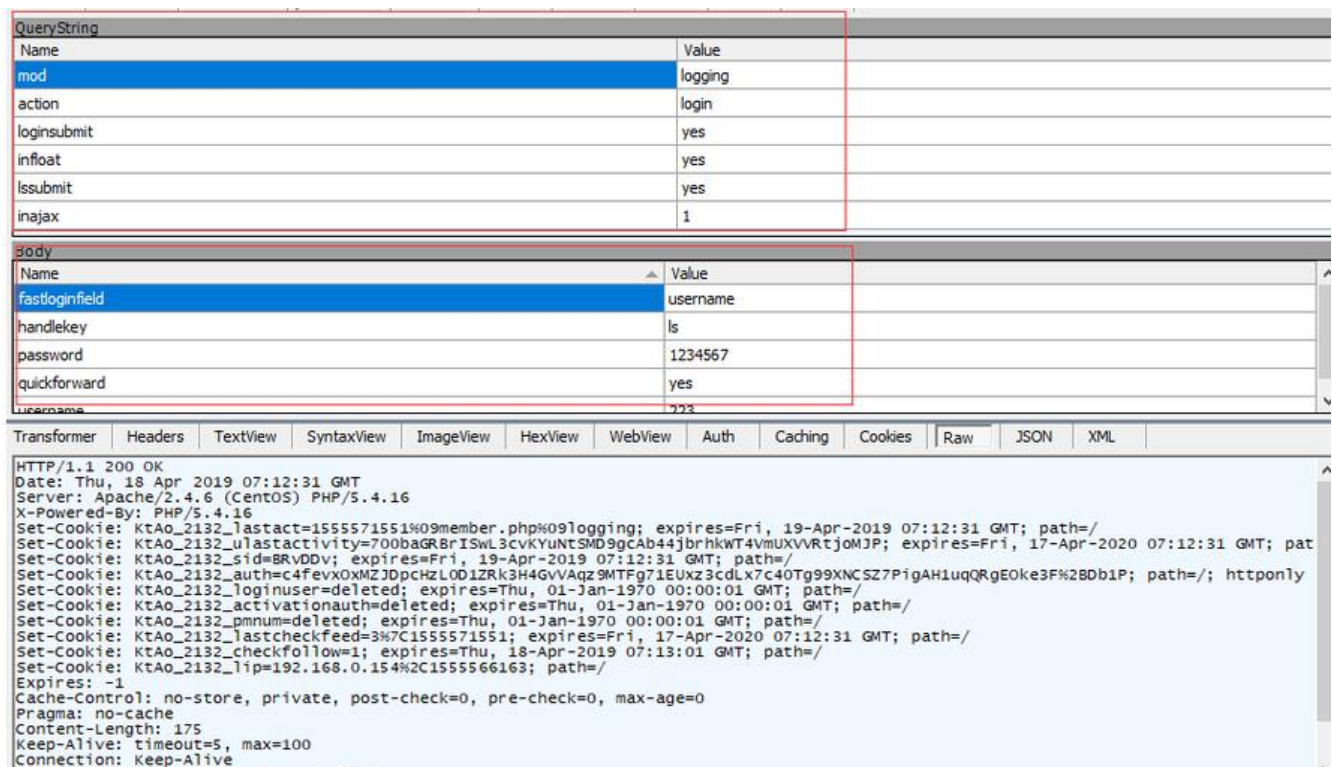
QueryString:
mod=logging
action=login
loginsubmit=yes
infloat=yes
lssubmit=yes
inajax=1

Body:
fastloginfield=username
username=223
password=1234567
quickforward=yes
handlekey=ls

Raw数据截图



WebForms数据截图



第二步：打开jmeter4.0,创建线程组和HTTP请求

创建线程组：TestPlan-->右击-->添加-->Threads(Users)-->线程组

创建HTTP请求：线程组-->右击-->添加-->Sampler-->HTTP请求

第三步：添加CSV数据文件设置

线程组-->右击-->添加-->配置元件-->CSV数据文件设置

第四步：对CSV数据文件设置页面进行设置

discuz论坛登录接口地址：<http://192.168.0.218/discuz/upload/member.php?mod=logging&action=login&loginsubmit=yes&infloat=yes&lssubmit=yes&inajax=1>

方法：POST接口

请求体：fastloginfield=username&username=223&password=1234567&quickforward=yes&andlekey=ls

进入CSV数据文件设置页面：

1 名称：可以默认

2 注释：可以为空

3 文件名：此处直接点击浏览，然后从指定路径下选择文件名以csv结尾的文件，如果不是，会报错在创建excel的时候，输入想要的参数后，不能直接改后缀，要在保存的时候选择文件保存格式，否则察看结果树会报错

打开csv文件（两列）：

username,password

223,12345676

224,1234567

225,1234567

226,1234567

227,1234567

228,1234567

4 文件编码：utf-8

5 变量名称（西文逗号间隔）:u,p

6 忽略首行（只在设置了变量名称后才生效）:True

7 分隔符：,

8 是否允许带?：False

9 遇到文件结束符将再一次循环：True

10 遇到文件结束符停止线程：False

11 线程共享：所有现场

Ps：其中7，8，9，10，11一般都选择默认

CSV 数据文件设置

名称: CSV 数据文件设置

注释:

设置 CSV 数据文件

文件名: C:\Users\brandal\Desktop\jmeter-link\username.csv 浏览...

文件编码: UTF-8

变量名称(西文逗号间隔): u,p

忽略首行(只在设置了变量名称后才生效): True

分隔符(用"\"代替制表符): ,

是否允许带引号?: False

遇到文件结束符再次循环?: True

遇到文件结束符停止线程?: False

线程共享模式: 所有现场

第五步：对HTTP请求页面进行设置

HTTP请求页面：

名称：discuz登录接口请求（此处可以默认不填）

注释：可以为空

协议：可以为空（默认是http协议）

服务器名称或ip：192.168.0.218

端口号：可以不填（默认80）（可以输入命令：vim /etc/httpd/conf/httpd.conf

在http.conf文件中可以查看到端口号）

方法：POST

路径：/discuz/upload/member.php

Content encoding:utf-8

点击Parameter，添加如下内容：

名称	值
mod	logging
action	login
loginsubmit	yes
infloat	yes
lssubmit	yes
inajax	1
fastloginfield	username
username	\${u}
password	\${p}
quickforward	yes
handlekey	ls

HTTP请求

名称:

注释:

Basic **Advanced**

Web服务器

协议: 服务器名称或IP: 端口号:

HTTP请求

方法: 路径: Content encoding:

☐ 自动重定向 ☒ 跟随重定向 ☒ Use KeepAlive ☐ Use multipart/form-data for POST ☐ Browser-compatible headers

Parameters **Body Data** **Files Upload**

同请求一起发送参数:

名称:	值	编码?	包含等于?
mod	logging	<input type="checkbox"/>	<input checked="" type="checkbox"/>
action	login	<input type="checkbox"/>	<input checked="" type="checkbox"/>
loginsubmit	yes	<input type="checkbox"/>	<input checked="" type="checkbox"/>
infloat	yes	<input type="checkbox"/>	<input checked="" type="checkbox"/>
lssubmit	yes	<input type="checkbox"/>	<input checked="" type="checkbox"/>
hajax	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
fastloginfield	username	<input type="checkbox"/>	<input checked="" type="checkbox"/>
username	\$(u)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
password	\$(p)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
quickforward	yes	<input type="checkbox"/>	<input checked="" type="checkbox"/>
handlekey	ls	<input type="checkbox"/>	<input checked="" type="checkbox"/>

第六步：添加集合点

笔者想实现的效果是模拟6个用户并发登录，如果不添加集合点，运行脚本的时候，数据是一条一条行，不存在并发，如果没有并发，就不会对接口形成压力

添加集合点：线程组-->右击-->添加-->定时器-->Synchronizing Timer

第七步：对集合点页面进行设置

在Synchronizing Timer页面：

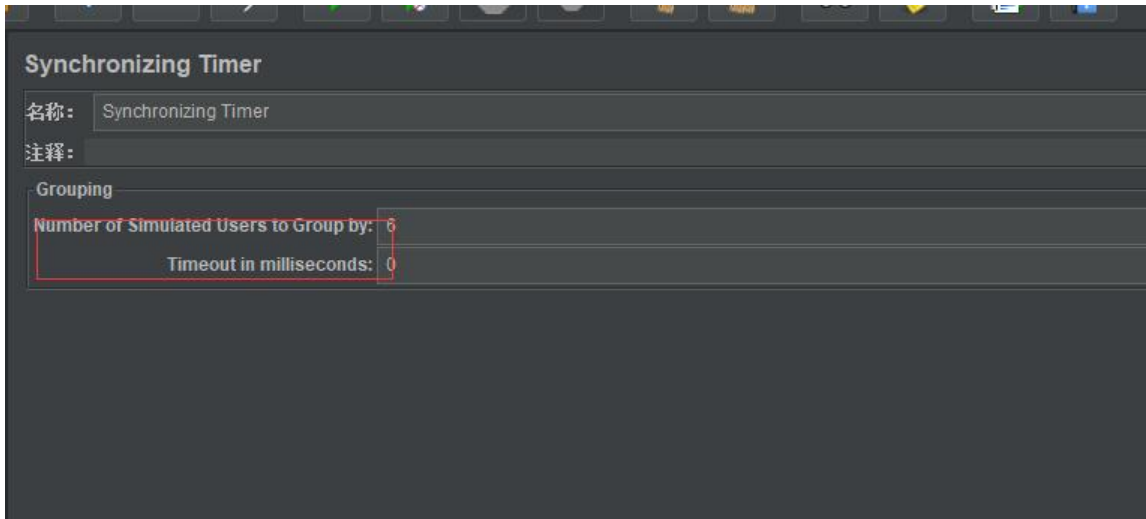
名称：可以默认

注释：可以为空

Number of simulated Users To Group by(集合多少个用户开始访问):6

Timeout in milliseconds: 0

注意，此处解释下Timeout in milliseconds:，如果设置为0，Timer将会等待线程数达到了"Number of Simultaneous Users to Group"中设置的值才释放。换句话说，如果线程数不足集合点中设置的数就会一直等待，需要手动stop。如果大于0，那么如果超过Timeout in milliseconds中设置的最大等待时间(毫秒为单位)后还没达到"Number of Simultaneous Users to Group"中设置的值，Timer将不等待，释放已到达的线程



第八步：添加查看结果树

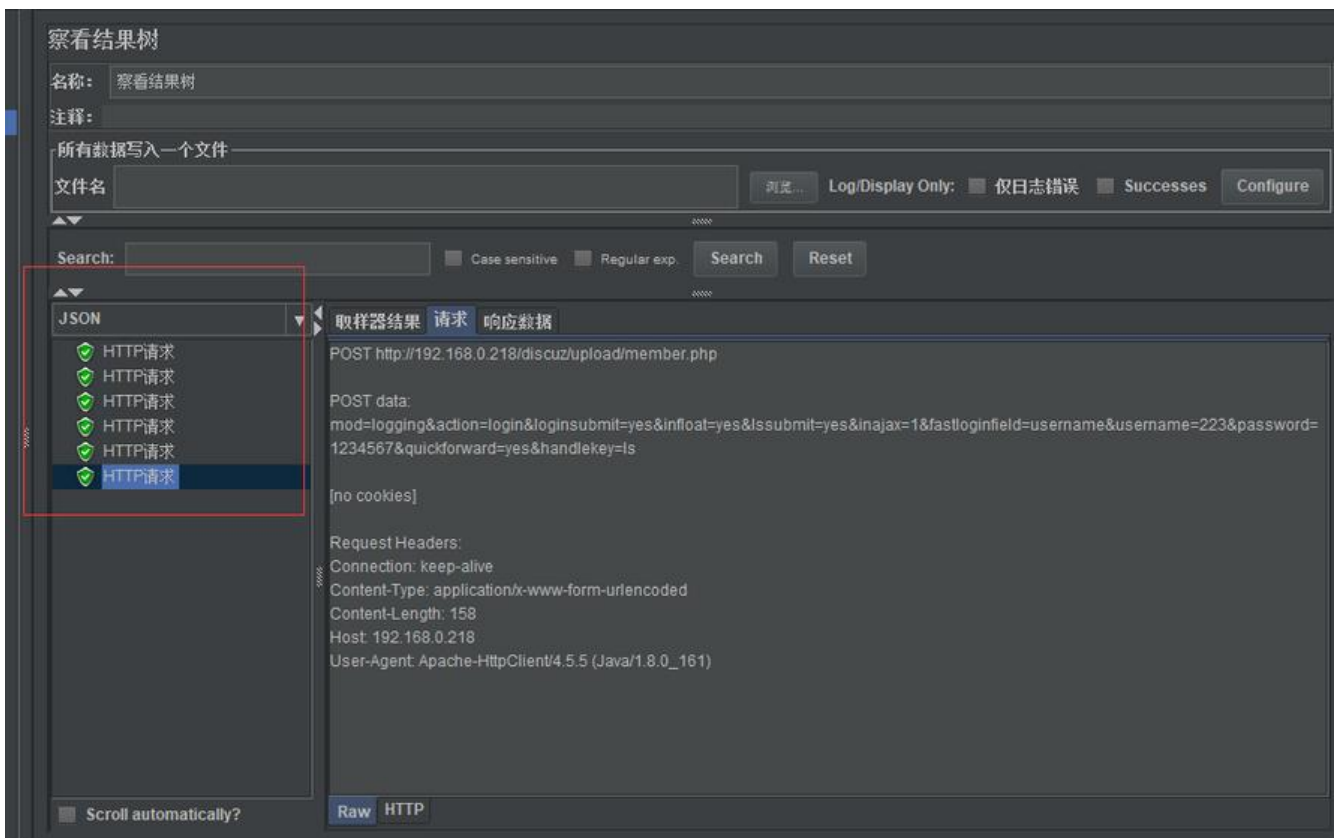
线程组-->右击-->添加-->监听器-->察看结果树

第九步：添加聚合报告

线程组-->右击-->添加-->监听器-->聚合报告

第九步：运行并查看结果

察看结果树结果



聚合报告结果

聚合报告

名称：

聚合报告

注释：

所有数据写入一个文件

文件名：

浏览...

Log/Display Only: ☐ 仅日志错误 ☐ Successes

Configure

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received K...	Sent KB/sec
HTTP请求	6	952	972	1005	1025	1025	869	1025	0.00%	5.8/sec	17.25	2.11
TOTAL	6	952	972	1005	1025	1025	869	1025	0.00%	5.8/sec	17.25	2.11

聚合报告字段解析

聚合报告参数解析：

- 1 sample：发出请求数量
- 2 Average：平均响应时间（默认单个请求的平均响应时间）
- 3 Median：中位数，50%用户的响应时间
- 4 90%Line：90%用户的响应时间
- 5 95%Line：95%用户的响应时间
- 6 99%Line：99%用户的响应时间
- 7 Min：最小响应时间
- 8 Max：最大响应时间
- Error%：本次测试中出现的错误的请求数量/请求的总数
- Throughout：吞吐量（每秒完成的请求数）
- KB/sec：每秒从服务器接收到的数据量

后记

当然，还有很多报告可以添加，比如系统资源监控的，个人可以根据实际情况添加。

以上，over

原创申明：以上内容都是根据笔者个人经验总结的，仅供参考，如需转载，请注明出处