

JMeter 接口/并发/性能测试（详尽版）

Jmeter 工具设计之初是用于做性能测试的，它在实现对各种接口的调用方面已经做的比较成熟，因此，本次直接使用 Jmeter 工具来完成对 Http 接口的测试。因为再做接口测试时可以设置线程组，所以也可做接口性能测试。

本篇使用 JMeter 完成了一个完整的 Http 接口性能测试流程，从脚本编写（工具化生成），执行到最终结果分析。

1. 测试接口的 基本要求

> 接口地址：我们选取最常见的百度搜索接口：

<https://www.baidu.com/s?ie=utf-8&&wd=新闻>

ie: 编码方式，默认为 utf-8

wd: 搜索词

> 返回结果判断 (断言):

搜索结果，我们可以通过校验结果中是否含有搜索词 **wd** 来判断本次请求成功或失败。

2. 操作

> 依次进行如下操作：1) 添加线程组 2)添加请求&http 消息头管理器
器 3)添加结果监听器（察看结果树）

4)添加用户自定义变量 *) 添加定时器（集合点）

5)添加请求的断言 6)添加断言结果 7) 添加聚合报告 8) 执行

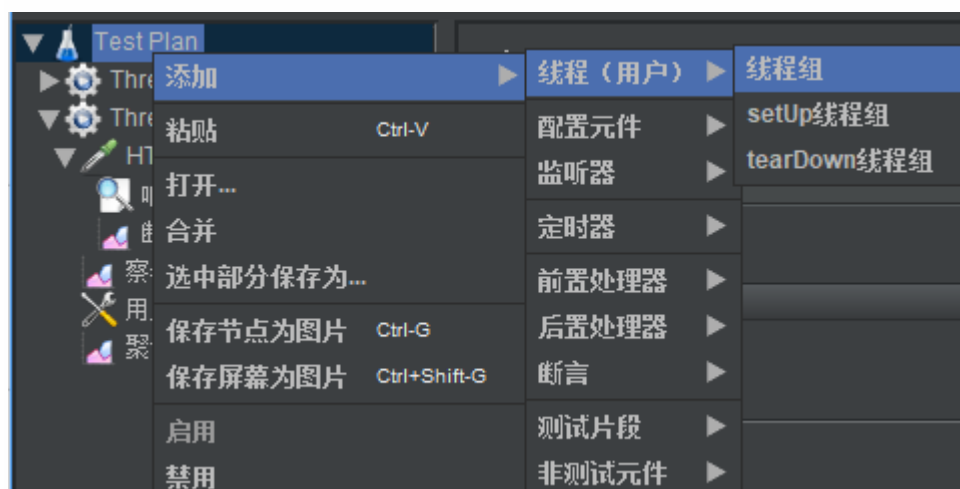
9) 结果分析（分析测试报告）

1) 添加线程组

添加虚拟用户组（Thread Group），这些虚拟用户用来模拟访问被测系统

一个 **线程组（Thread Group）** 等于 **一组虚拟用户（virtual users group）**，

一个 Java 线程 等于 一个用户



1) 添加线程组 --菜单



1) 添加线程组 --参数

线程组参数详解：

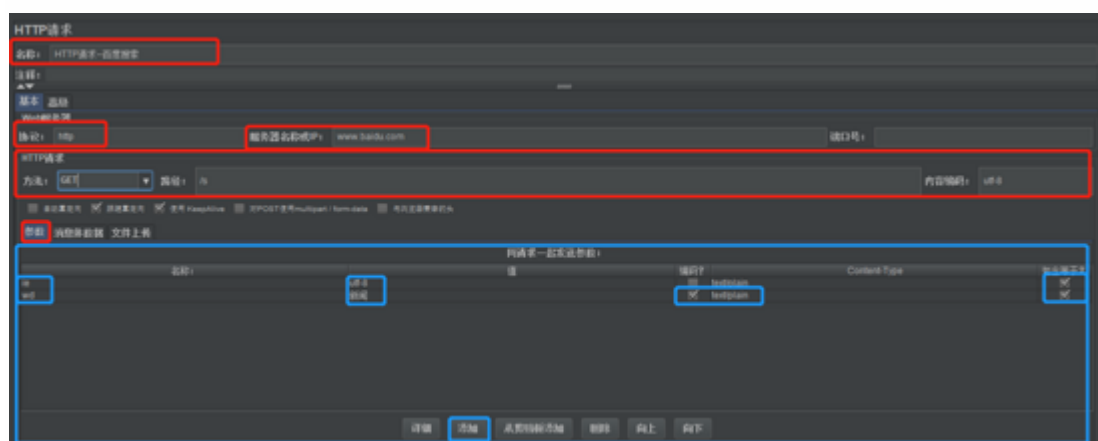
1. 线程数：虚拟用户数。一个虚拟用户占用一个进程或线程。设置多少虚拟用户数在这里也就是设置多少个线程数。
 2. Ramp-Up Period(in seconds)准备时长：设置的虚拟用户数需要多长时间全部启动。如果线程数为10，准备时长为2，那么需要2秒钟启动10个线程，也就是每秒钟启动5个线程。
 3. 循环次数：每个线程发送请求的次数。如果线程数为10，循环次数为100，那么每个线程发送100次请求。总请求数为10*100=1000。如果勾选了“永远”，那么所有线程会一直发送请求，一到选择停止运行脚本。
 4. Delay Thread creation until needed：直到需要时延迟线程的创建。
 5. 调度器：设置线程组启动的开始时间和结束时间(配置调度器时，需要勾选循环次数为永远)
- 持续时间 (秒)：测试持续时间，会覆盖结束时间
启动延迟 (秒)：测试延迟启动时间，会覆盖启动时间
启动时间：测试启动时间，启动延迟会覆盖它。当启动时间已过，手动只需测试时当前时间也会覆盖它。
结束时间：测试结束时间，持续时间会覆盖它。

2)添加请求、添加 http 消息头管理器

2.1)添加请求

接下来的一步就是要加入实际被测页面，右击“线程组”>“添加”>“Sampler”>“HTTP 请求”。

ps: 参数，用于 get 或 put 请求；消息体数据用于 post 请求；-----很重要，很容易出错！



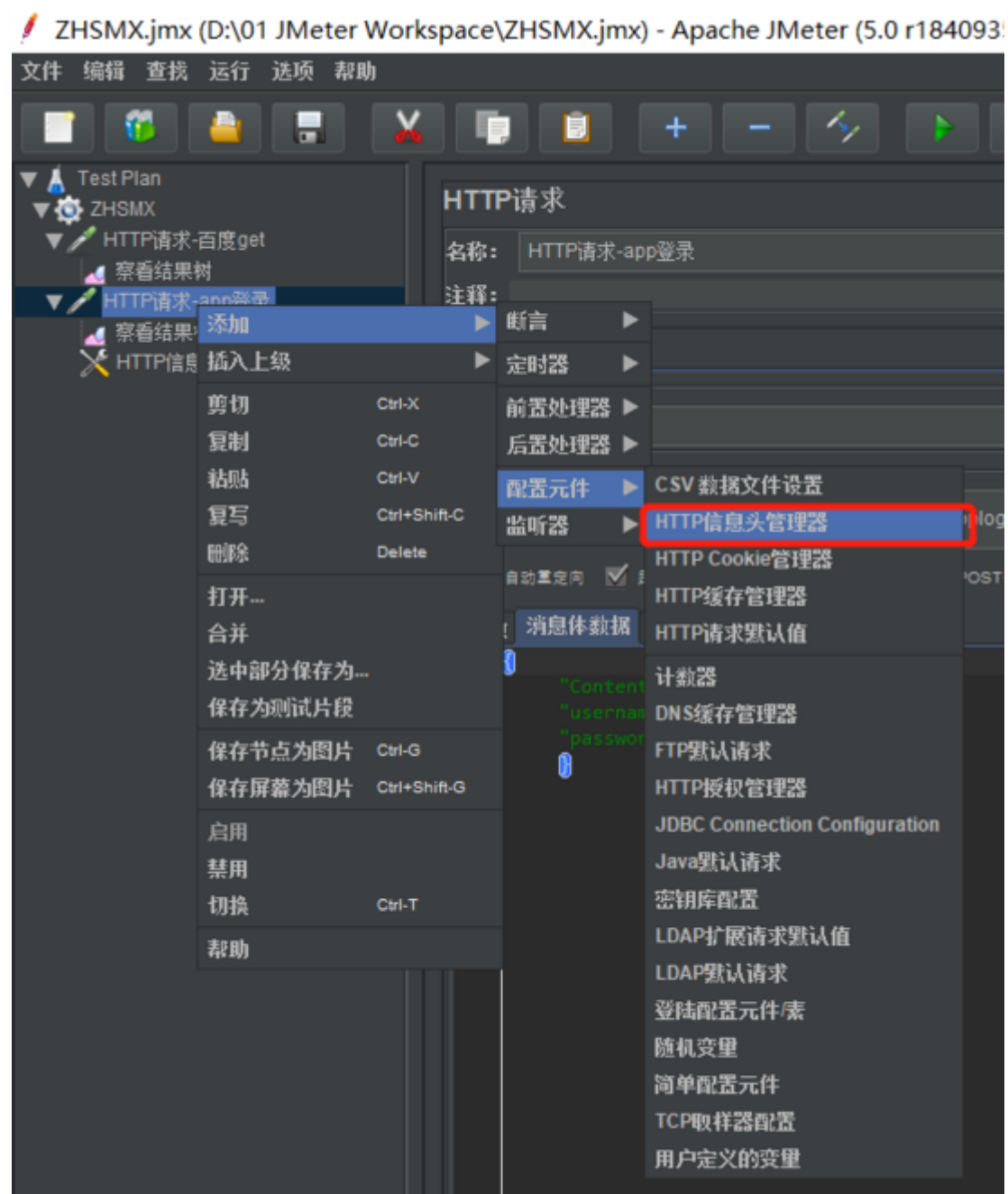
1) 添加请求

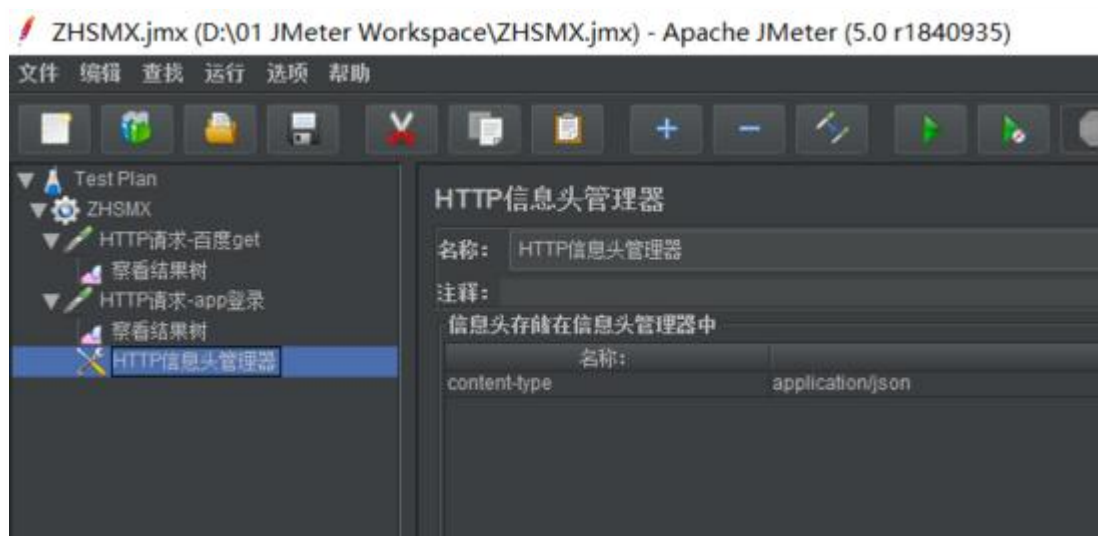
Http请求主要参数详解：

1. Web服务器
协议：向目标服务器发送HTTP请求协议，可以是HTTP或HTTPS，默认为HTTP
服务器名称或IP：HTTP请求发送的目标服务器名称或IP
端口号：目标服务器的端口号，默认值为80
2. Http请求
方法：发送HTTP请求的方法，可用方法包括GET、POST、HEAD、PUT、OPTIONS、TRACE、DELETE等。
路径：目标URL路径（URL中去掉服务器地址、端口及参数后剩余部分）
Content encoding：编码方式，默认为ISO-8859-1编码，这里配置为utf-8
2. 同请求一起发送参数
在请求中发送的URL参数。用户可以将URL中所有参数设置在本表中，表中每行为一个参数（对应URL中的 name=value），注意参数传入中文时需要勾选“编码”

上图注释

2.2)http 消息头管理器

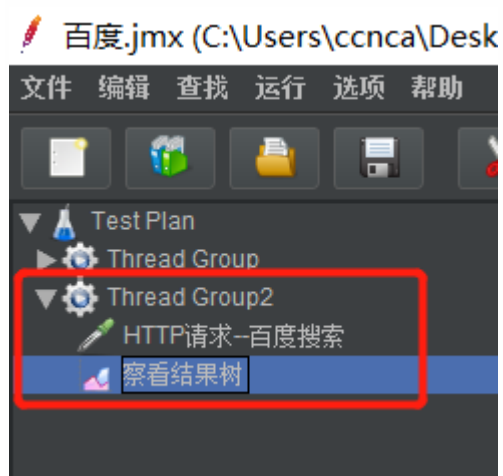




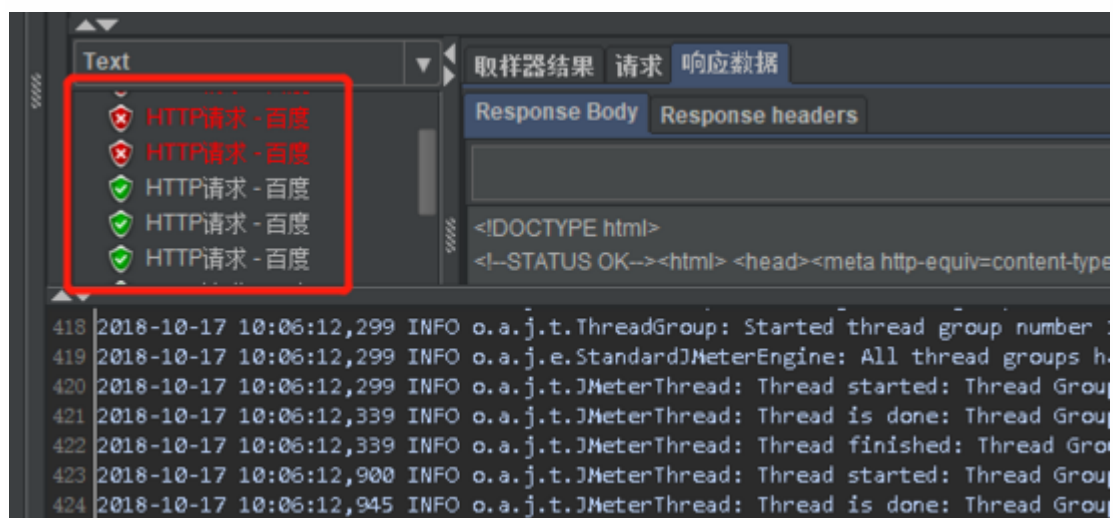
3) 添加察看结果树

添加结果监听器（察看结果树），“监听器”让用户来观察测试结果

运行后点击左侧观察结果数，查看请求执行结果，红色为失败，绿色为成功，如下图：

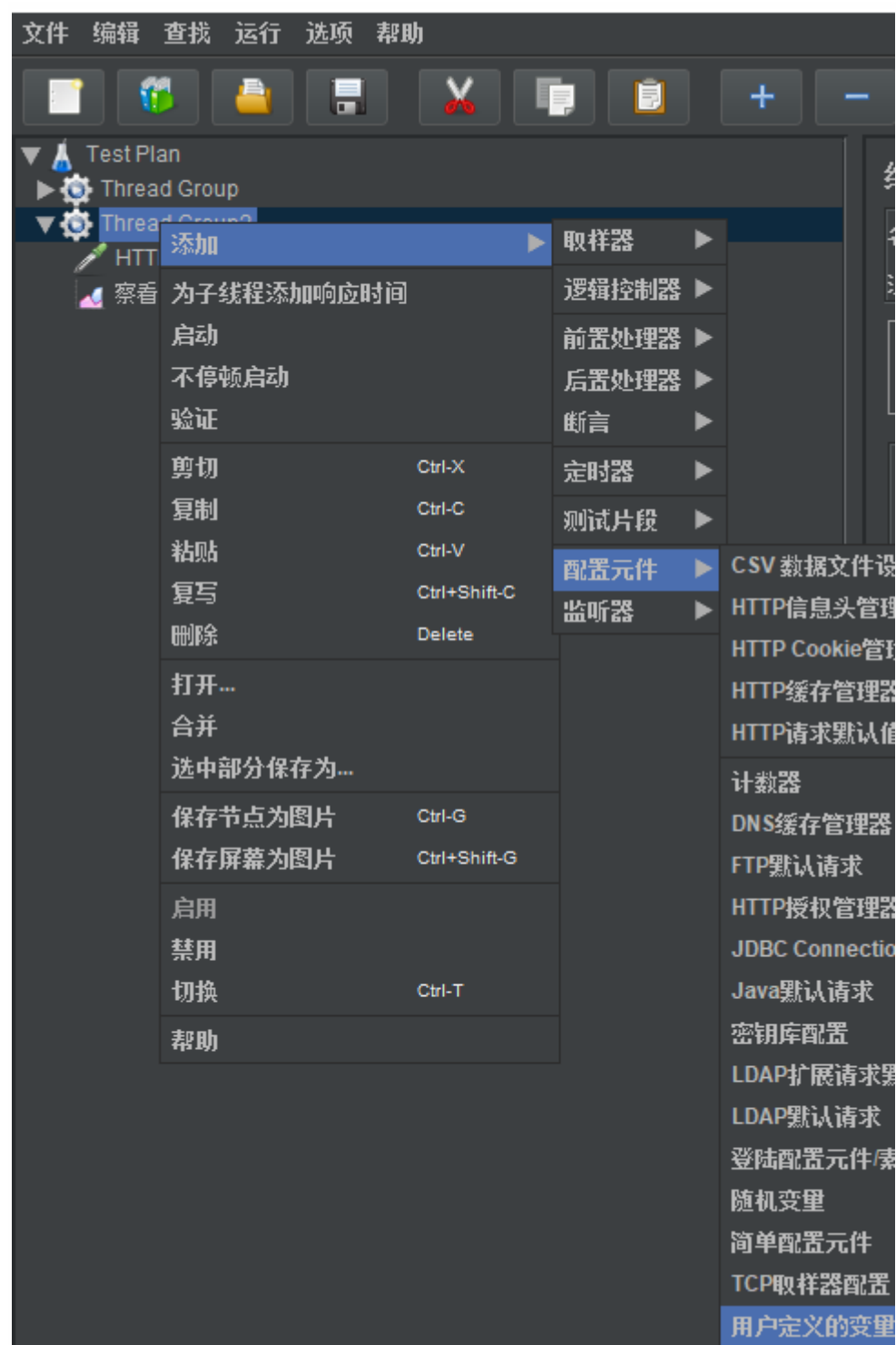


3) 添加察看结果树

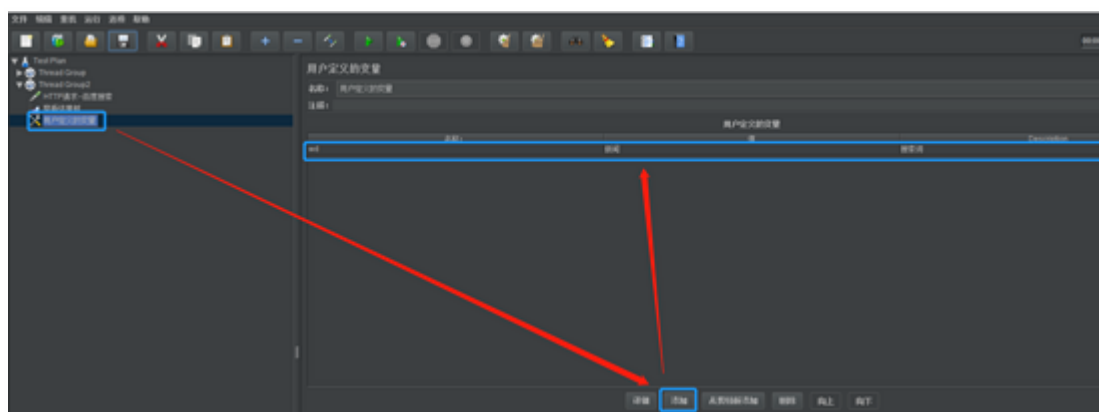


执行后结果树查看

4)添加用户自定义变量 &集合点

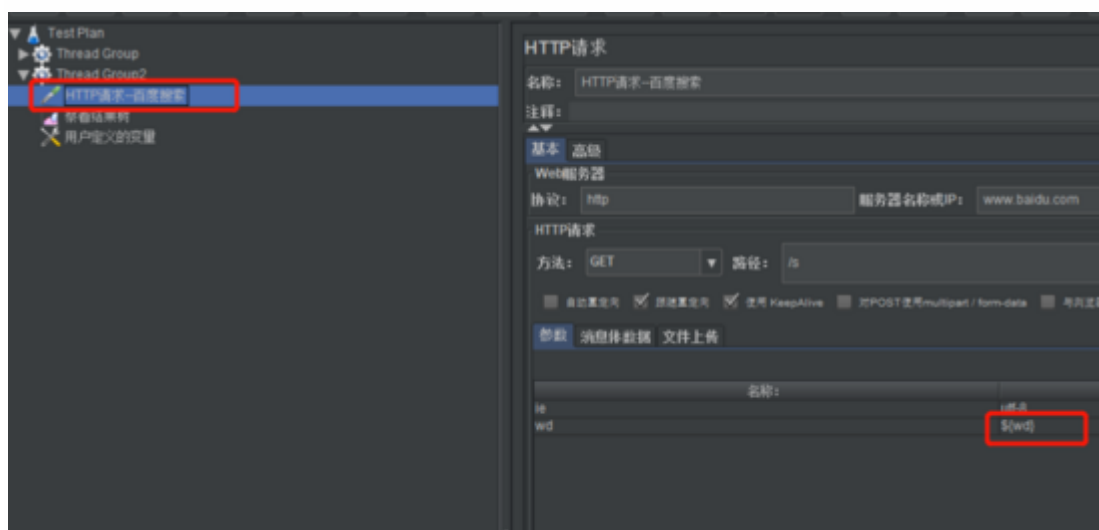


4)添加用户自定义变量 -菜单

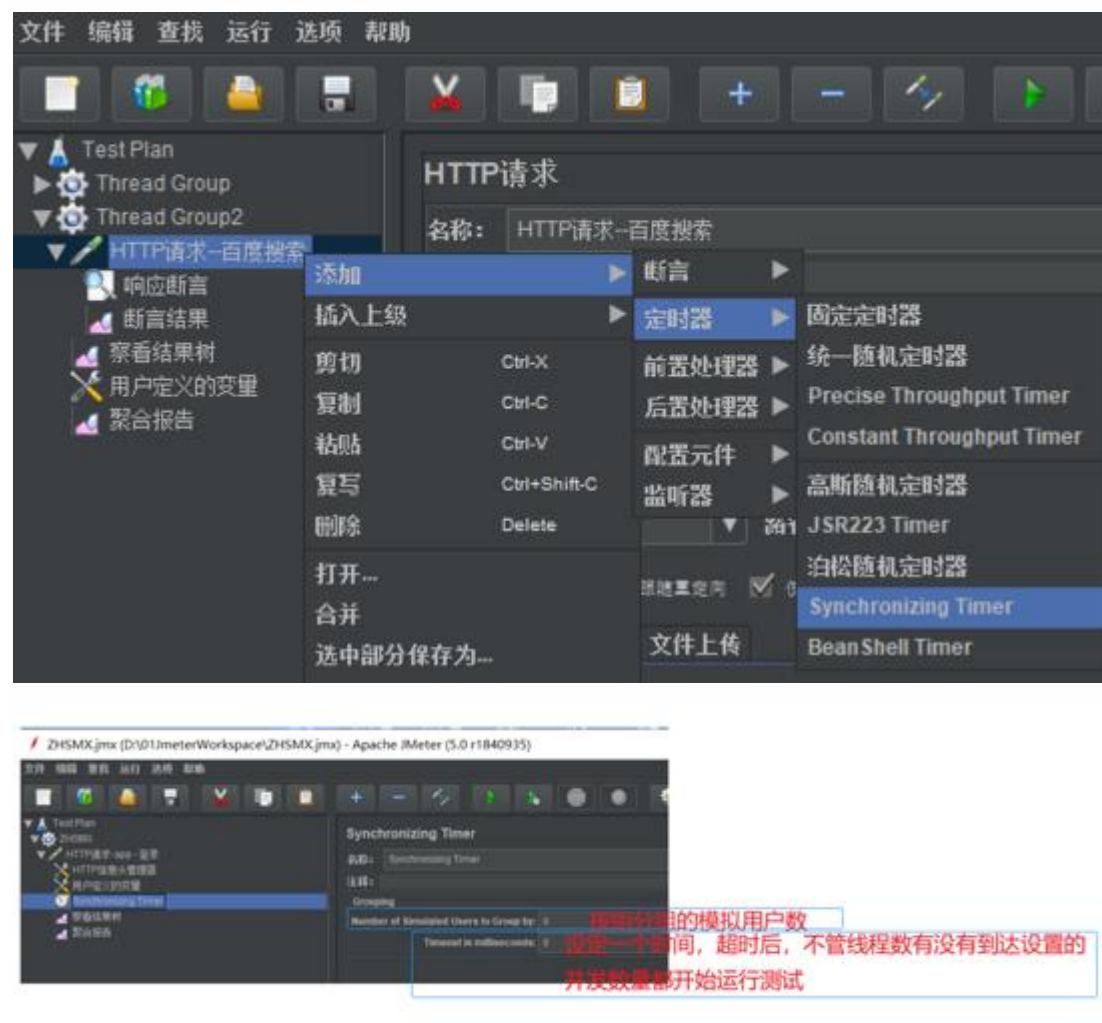


4) 添加用户自定义变量 - 添加变量

用户自定义变量 -, 需要在 HTTP 请求处, 手动修改请求中为变量形式 `${wd}`



*) 添加集合点



如果线程数是 6，按组分组的模拟用户数是 3，超时时间是 0，则：当加载完 3 个用户时，这 3 个用户开始并发操作；当另外 3 个用户加载完时，开始第 2（6/3）组的并发；-----如果要做 1000 用户并发，则第一个参数为 1000，第二个参数为 0。

结论：

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。也就是说，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 将不再等待，释放已到达的线程。也就是说如果线程数不满足集合点中设置的值，则在 timeout 中设置的时间后继续执行不足的那些线程。

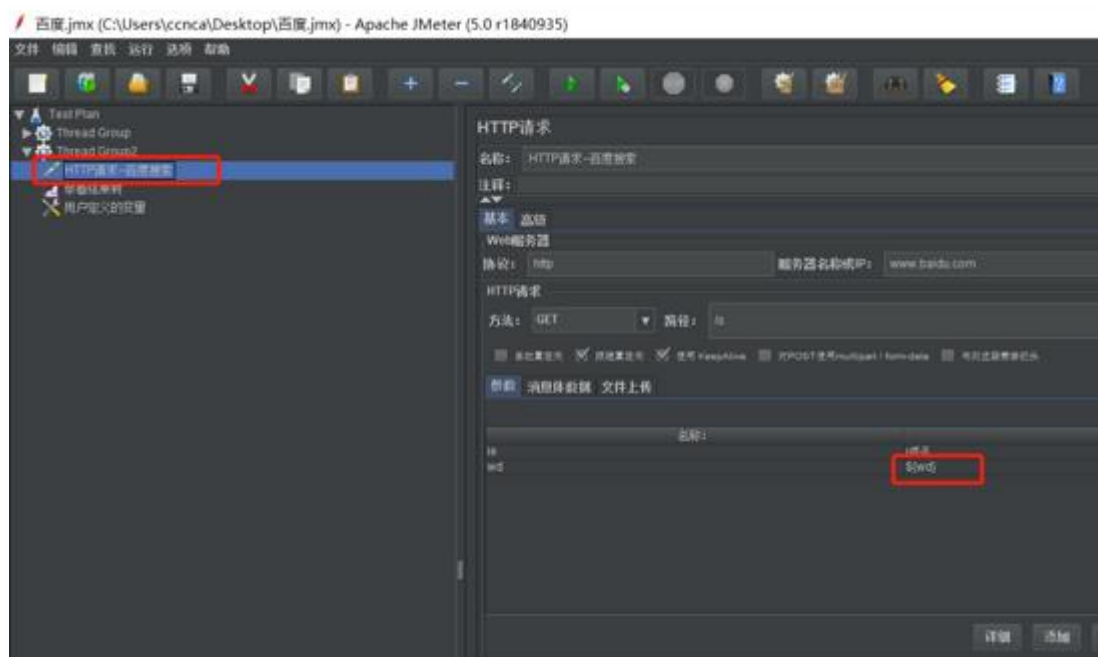
Timeout in milliseconds 默认为 0。所以当 timeout 设置为 0，但是线程数又不满足集合点中设置的值时，就会一直等待，不执行请求，需要手动 stop。

同步定时器是在每一个采集器之前执行的，不管定时器的位置是在采集器之前还是之后，都是在采集器之前执行。

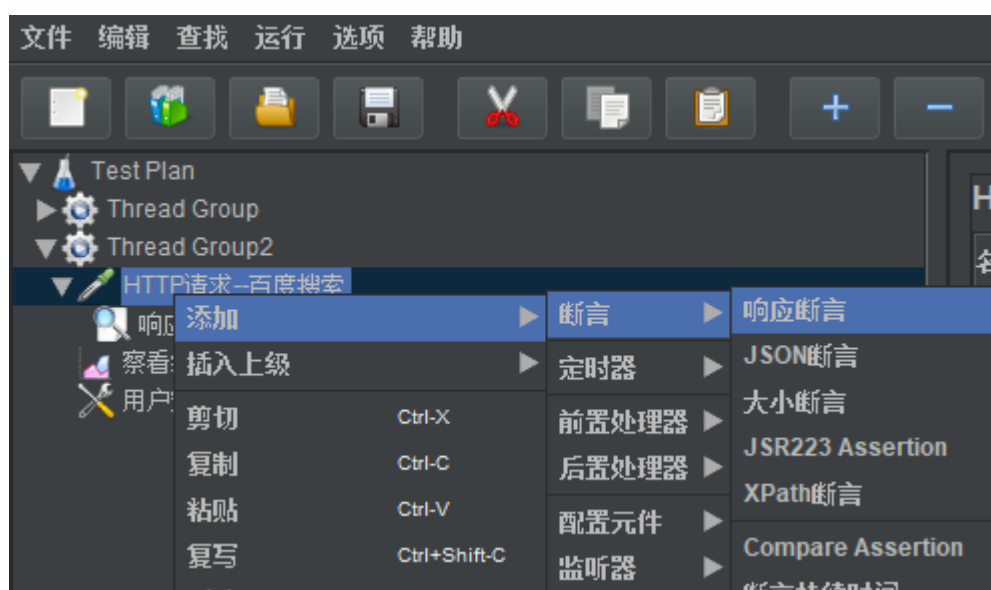
如果一个线程中存在多个采集器，同步定时器和这些采集器在同一级（同一节点下），则同时作用于这些采集器。

如果需要有一个定时器单独对应某一个采集器，可以在采集器的子节点中创建定时器。

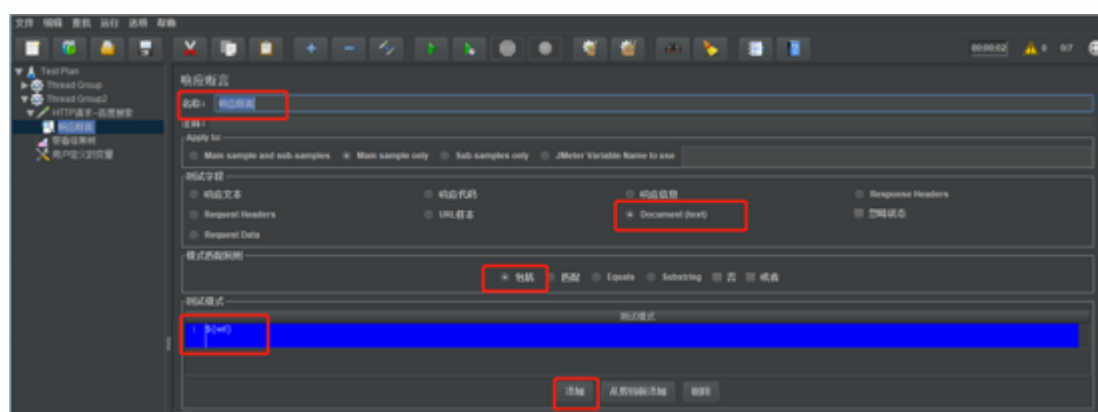
5) 添加请求的断言 6) 添加断言结果



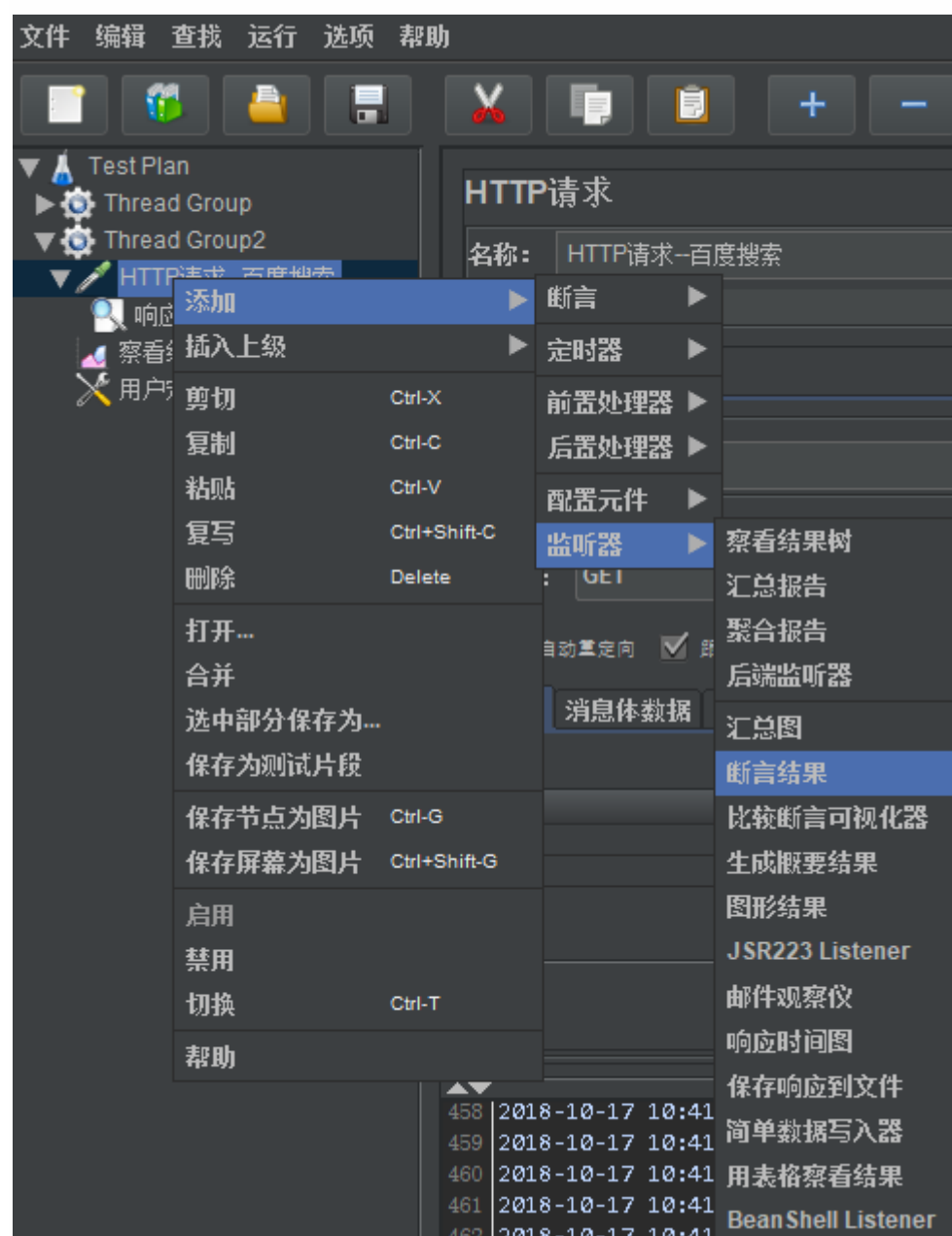
4) 添加用户自定义变量 - 手动修改请求中为变量形式 $\{wd\}$



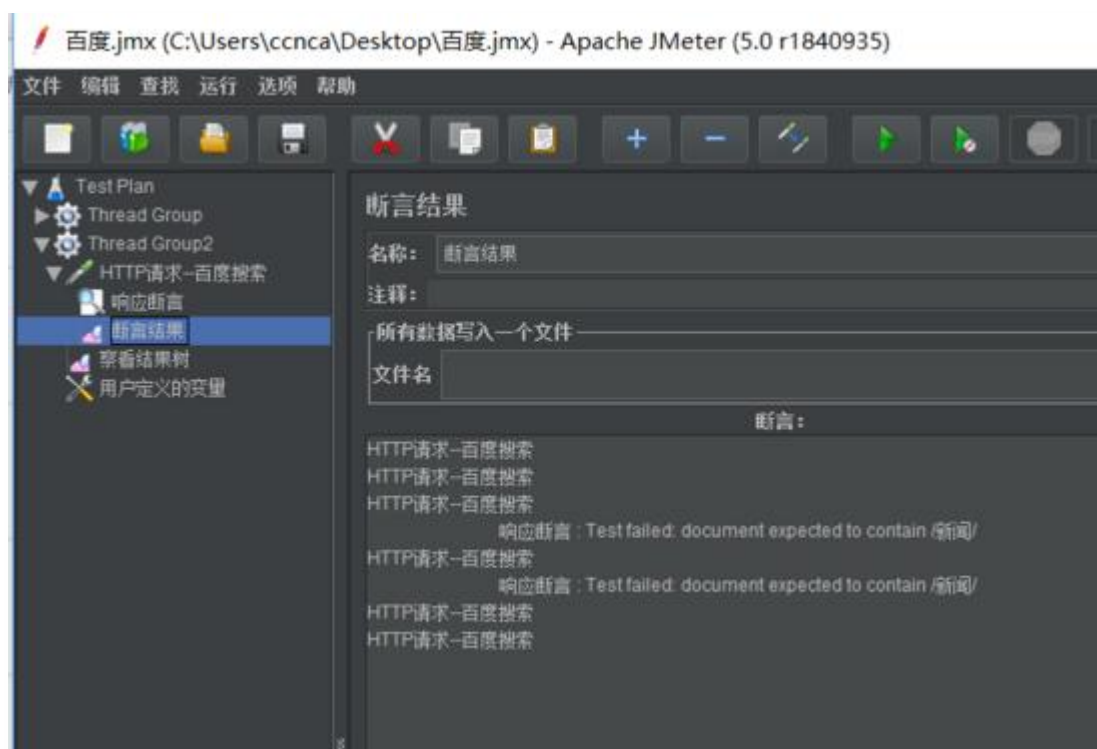
5) 添加请求的断言 -- 菜单



5) 添加请求的断言 -- 校验返回的文本中是否包含搜索词, 添加参数`${wd}`到要测试的模式中

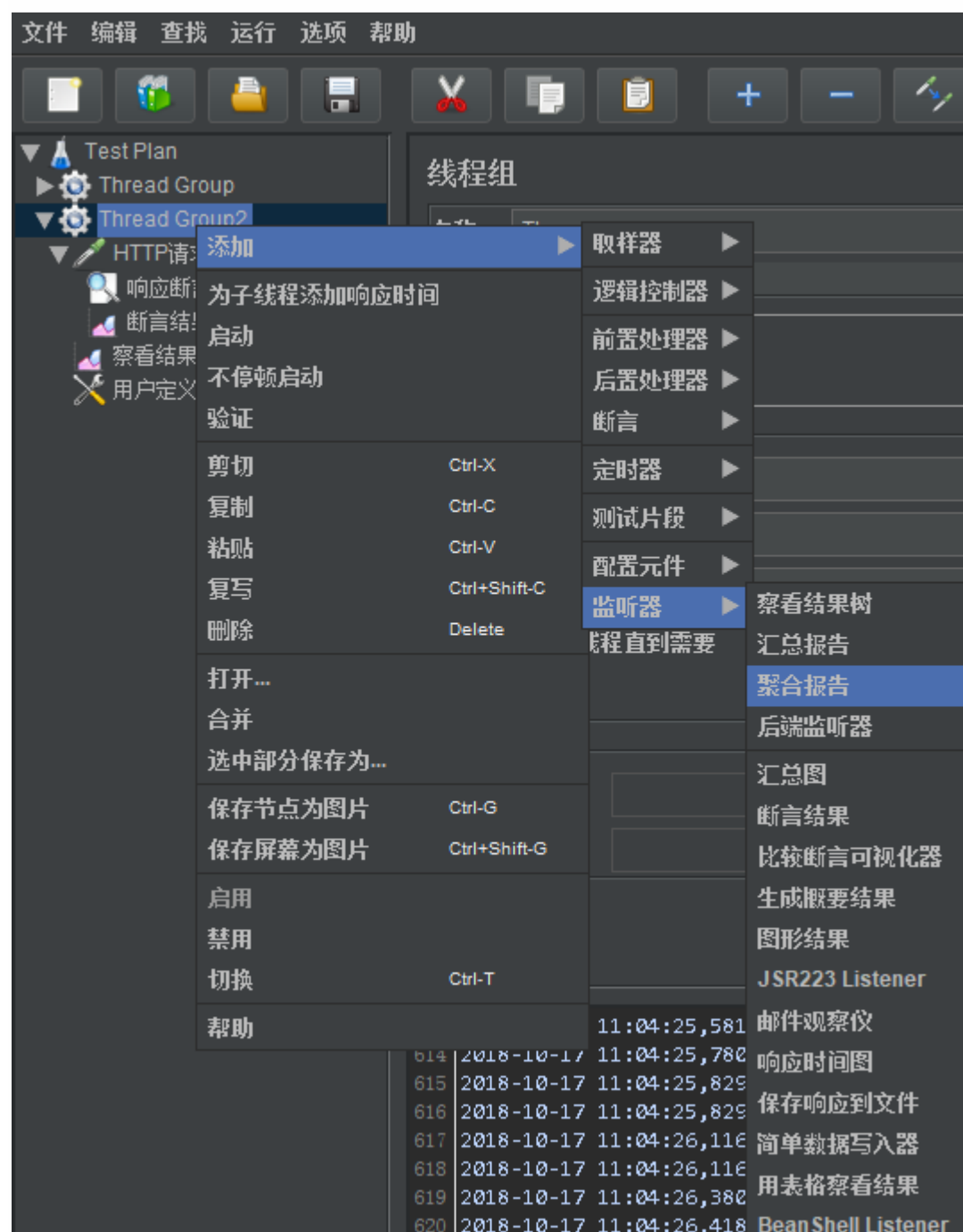


6) 添加断言结果 --菜单, 运行一次就可以看到断言结果成功或失败了



6) 断言结果查看，如上失败，可能是添加断言的“测试字段”勾选错误导致

7) 添加聚合报告



7) 添加聚合报告--“线程组”->“添加”->“监听器”->“聚合报告”，用以存放性能测试报告

8) 执行

至此，就完成了完整 Http 接口的 JMeter 性能测试脚本编写，点击运行即可

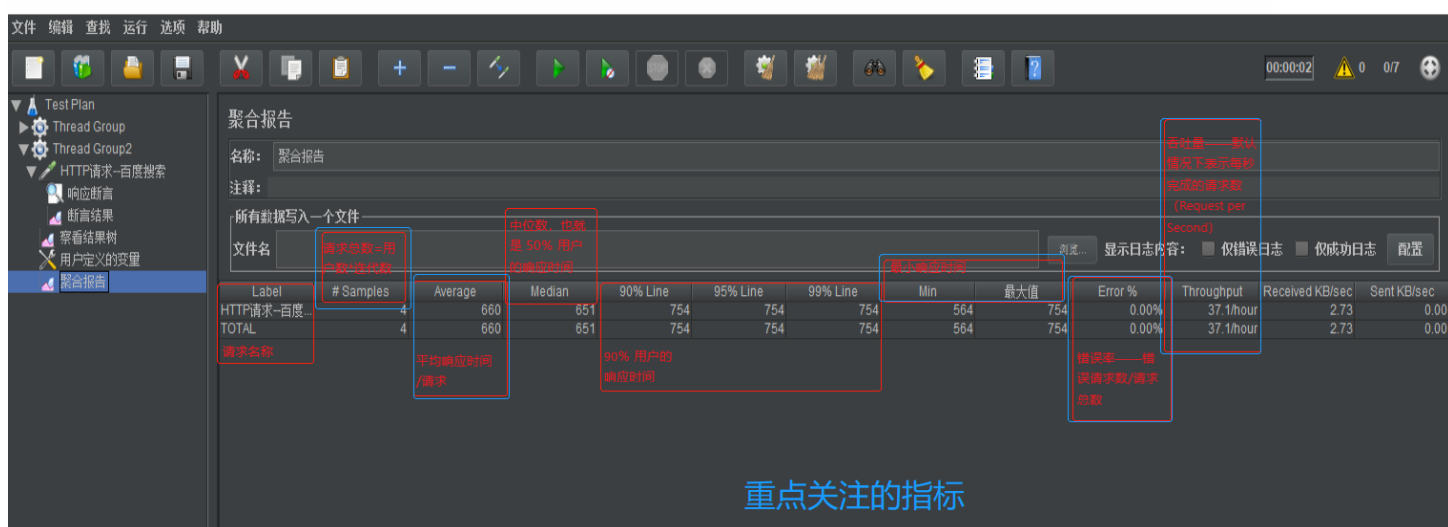
>查看运行日志，“选项”>“Log Viewer”看看运行的日志。如果你还是看不到日志，你可以点击下面所示的三角箭头展开或者收起日志视图



8) 运行 （后面是清空执行结果）

9) 结果分析（分析测试报告）

待性能测试执行完成后，打开聚合报告可以看到：



响应时间的单位是 ms. 另外：KB/Sec: 每秒从服务器端接收到的数据量，相当于 LoadRunner 中的 Throughput/Sec

10) 脚本查看

可以使用编辑器查看.jmx 文件，如下图所示：

C:\Users\cnca\Desktop\百度.jmx - Notepad++

文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(I) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?

```
91 <stringProp name="ThreadGroup.num_threads">2</stringProp>
92 <stringProp name="ThreadGroup.ramp_time">2</stringProp>
93 <boolProp name="ThreadGroup.scheduler">false</boolProp>
94 <stringProp name="ThreadGroup.duration"></stringProp>
95 <stringProp name="ThreadGroup.delay"></stringProp>
96 </ThreadGroup>
97 <hashTree>
98 <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="HTTP请求--百度搜索" enabled="true">
99 <elementProp name="HTTPSampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" testname="用户定义的变量" enabled="true">
100 <collectionProp name="Arguments.arguments">
101 <elementProp name="ie" elementType="HTTPArgument">
102 <boolProp name="HTTPArgument.always_encode">false</boolProp>
103 <stringProp name="Argument.value">utf-8</stringProp>
104 <stringProp name="Argument.metadata">=</stringProp>
105 <boolProp name="HTTPArgument.use_equals">true</boolProp>
106 <stringProp name="Argument.name">ie</stringProp>
107 </elementProp>
108 <elementProp name="wd" elementType="HTTPArgument">
109 <boolProp name="HTTPArgument.always_encode">true</boolProp>
110 <stringProp name="Argument.value">${wd}</stringProp>
111 <stringProp name="Argument.metadata">=</stringProp>
112 <boolProp name="HTTPArgument.use_equals">true</boolProp>
113 <stringProp name="Argument.name">wd</stringProp>
114 </elementProp>
115 </collectionProp>
116 </elementProp>
117 <stringProp name="HTTPSampler.domain">www.baidu.com</stringProp>
118 <stringProp name="HTTPSampler.port"></stringProp>
119 <stringProp name="HTTPSampler.protocol">http</stringProp>
120 <stringProp name="HTTPSampler.contentEncoding">utf-8</stringProp>
121 <stringProp name="HTTPSampler.path">/s</stringProp>
122 <stringProp name="HTTPSampler.method">GET</stringProp>
123 <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
124 <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
125 <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
126 <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
127 <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
128 <stringProp name="HTTPSampler.connect_timeout"></stringProp>
129 <stringProp name="HTTPSampler.response_timeout"></stringProp>
130 </HTTPSamplerProxy>
131 <hashTree>
132 <ResponseAssertion guiclass="AssertionGui" testclass="ResponseAssertion" testname="响应断言" enabled="true">
133 <collectionProp name="Assertion.test_strings">
134 <stringProp name="37028633">${wd}</stringProp>
135 </collectionProp>
136 <stringProp name="Assertion.custom_message"></stringProp>
137 <stringProp name="Assertion.test_field">Assertion.response_data_as_document</stringProp>
138 <boolProp name="Assertion.assume_success">false</boolProp>
```