

使用 JMeter 进行压力测试

一.前言

压力测试是每一个Web应用程序上线之前都需要做的一个测试，他可以帮助我们发现系统中的瓶颈问题，减少发布到生产环境后出问题的几率；预估系统的承载能力，使我们能根据其做出一些应对措施。所以压力测试是一个非常重要的步骤，下面我带大家来使用一款压力测试工具JMeter。

二.关于JMeter

Apache JMeter是Apache组织开发的基于Java的压力测试工具。用于对软件做压力测试，它最初被设计用于Web应用测试，但后来扩展到其他测试领域。它可以用于测试静态和动态资源，例如静态文件、Java 小服务程序、CGI 脚本、Java 对象、数据库、FTP 服务器，等等。JMeter 可以用于对服务器、网络或对象模拟巨大的负载，来自不同压力类别下测试它们的强度和分析整体性能。另外，JMeter能够对应用程序做功能/回归测试，通过创建带有断言的脚本来验证你的程序返回了你期望的结果。为了最大限度的灵活性，JMeter允许使用正则表达式创建断言。

Apache jmeter 可以用于对静态的和动态的资源（文件，Servlet，Perl脚本，java 对象，数据库和查询，FTP服务器等等）的性能进行测试。它可以用于对服务器、网络或对象模拟繁重的负载来测试它们的强度或分析不同压力类型下的整体性能。你可以使用它做性能的图形分析或在大并发负载测试你的服务器/脚本/对象。

官网：http://jmeter.apache.org/download_jmeter.cgi

这里我选用了 4.0 版本的二进制包：<http://mirror.bit.edu.cn/apache//jmeter/binaries/apache-jmeter-4.0.zip>

更多内容介绍 <https://baike.baidu.com/item/Jmeter/3104456>

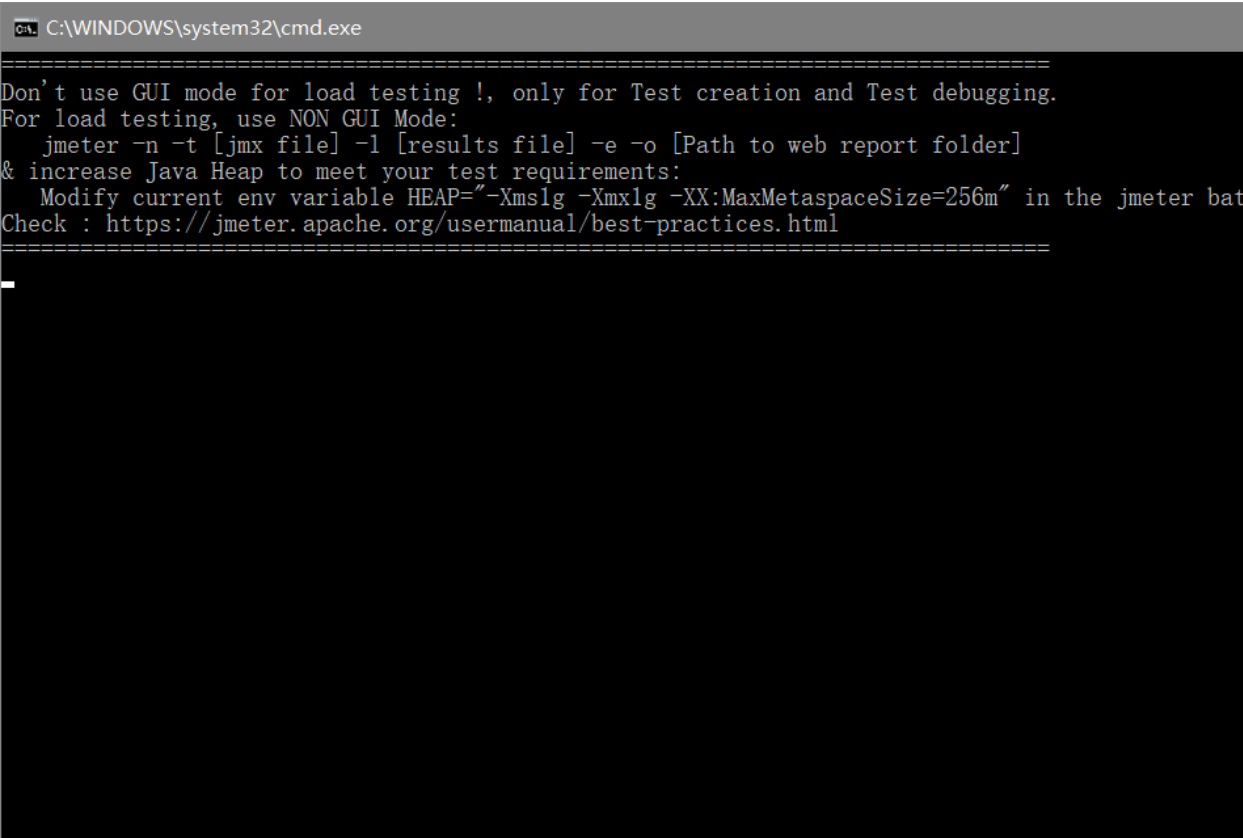
三.准备工作

因为JMeter是使用JAVA写的，所以使用JMeter之前，先安装JAVA环境，本文就不讲不如安装JAVA环境了。.

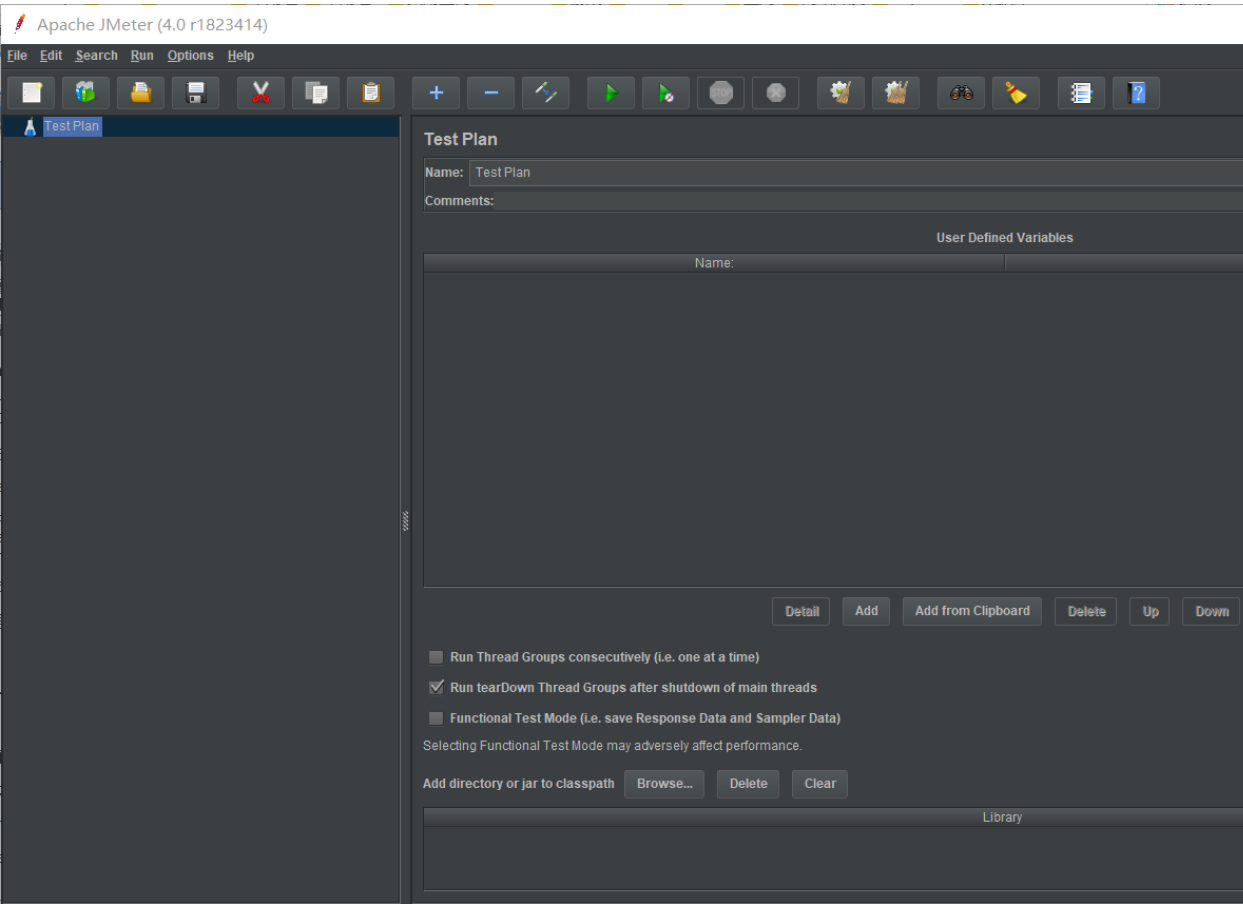
JAVA环境变量配置：<https://jingyan.baidu.com/article/fd8044fa2c22f15031137a2a.html>

解压下载的二进制包，进入bin目录，使用 `jmeter.bat` 启动程序。

启动之后会有两个窗口，一个cmd窗口，一个JMeter的 GUI。前面**不要忽略**CMD窗口的提示信息：



JMeter:



CMD窗口的提示信息

```
=====
Don't use GUI mode for load testing !, only for Test creation and Test debugging
```

For load testing, use NON GUI Mode:

```
jmeter -n -t [jmx file] -l [results file] -e -o [Path to web report folder]
& increase Java Heap to meet your test requirements:
Modify current env variable HEAP="-Xms1g -Xmx1g -XX:MaxMetaspaceSize=256m" i
Check : https://jmeter.apache.org/usermanual/best-practices.html
```

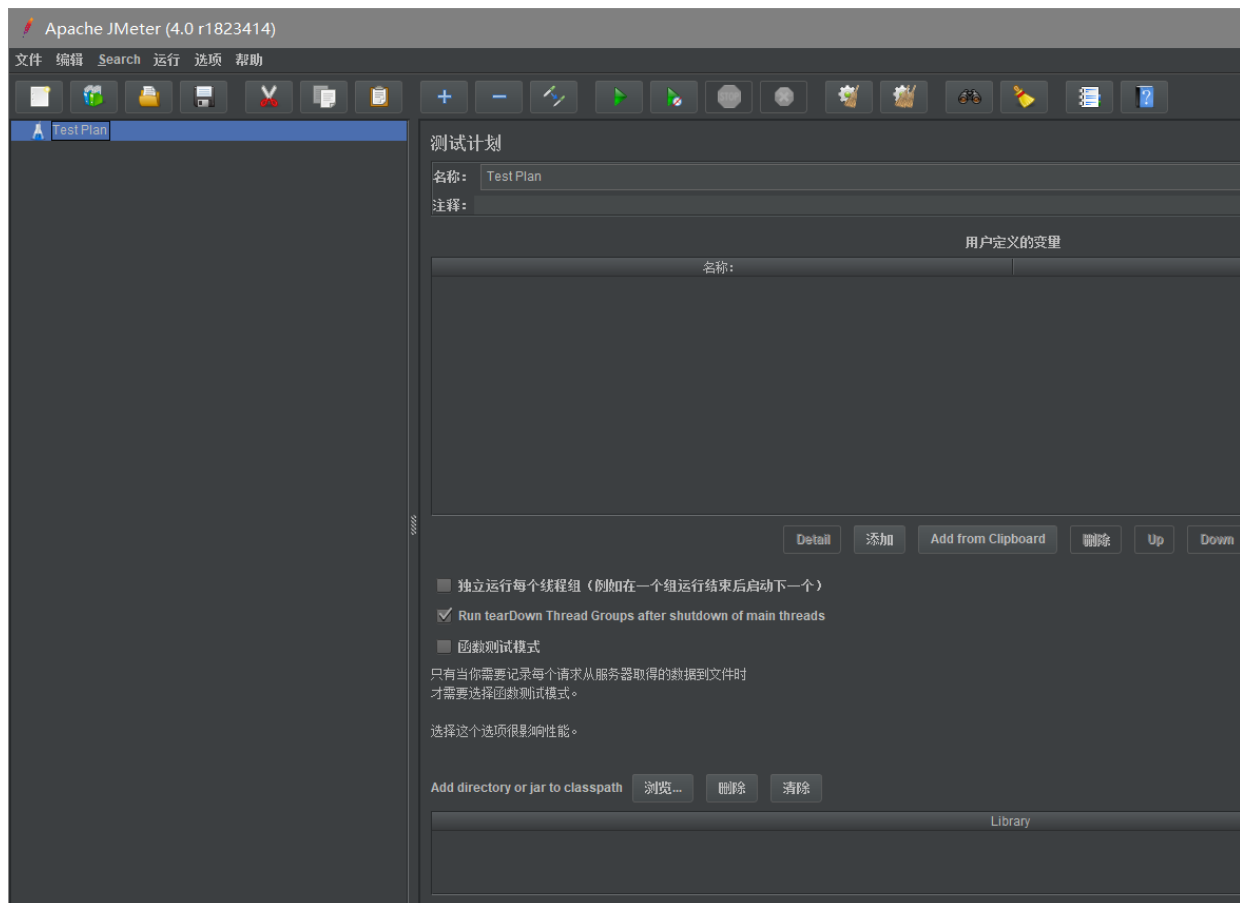
上面的意思就是：不要使用GUI运行压力测试，GUI仅用于压力测试的创建和调试；执行压力测试请不要使用GUI。使用下面的命令来执行测试：

```
jmeter -n -t [jmx file] -l [results file] -e -o [Path to web report folder]
```

并且修改JMeter批处理文件的环境变量：`HEAP="-Xms1g -Xmx1g -XX:MaxMetaspaceSize=256m"`

更改语言为中文

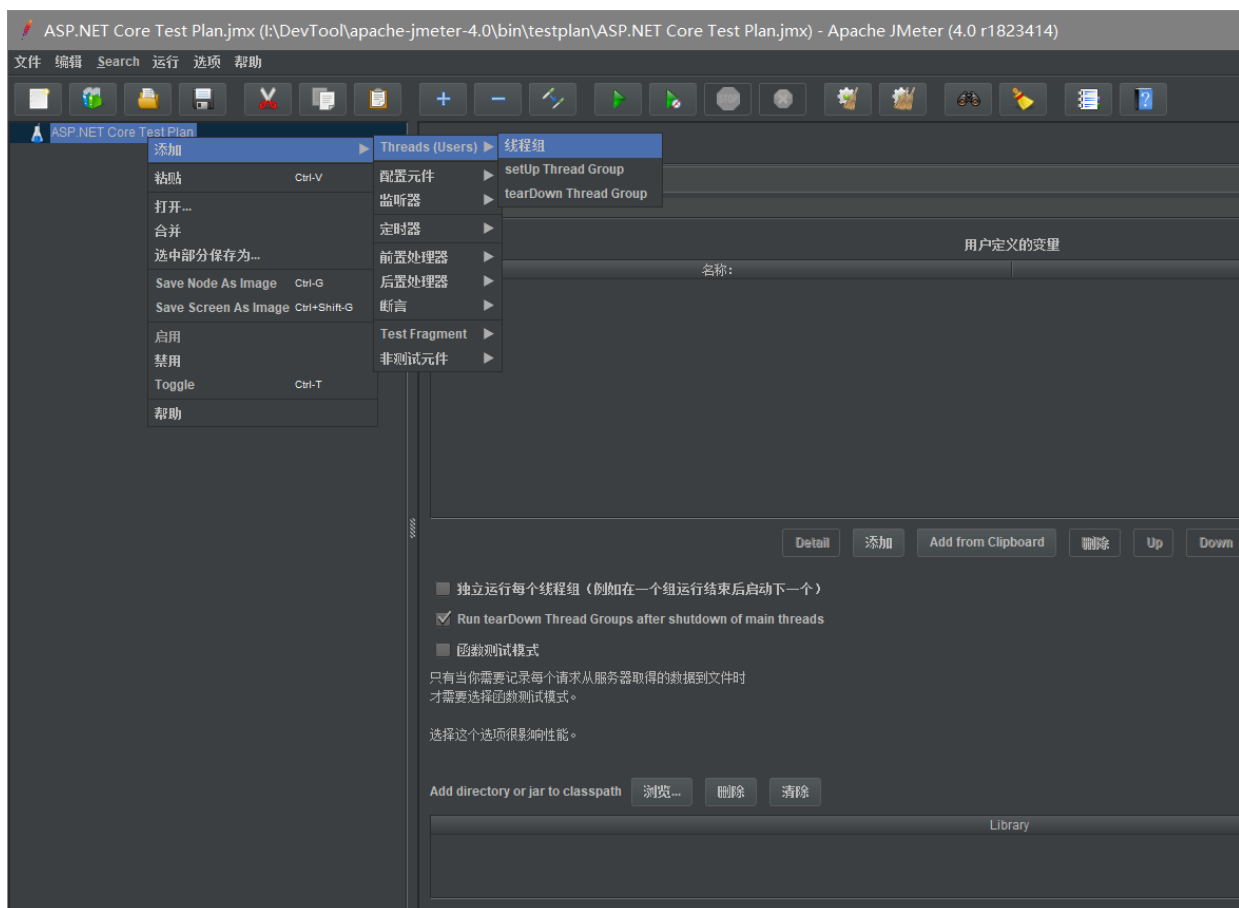
官方默认为我们提供了简体中文。通过【Options】->【Choose Language】变更为简体中文



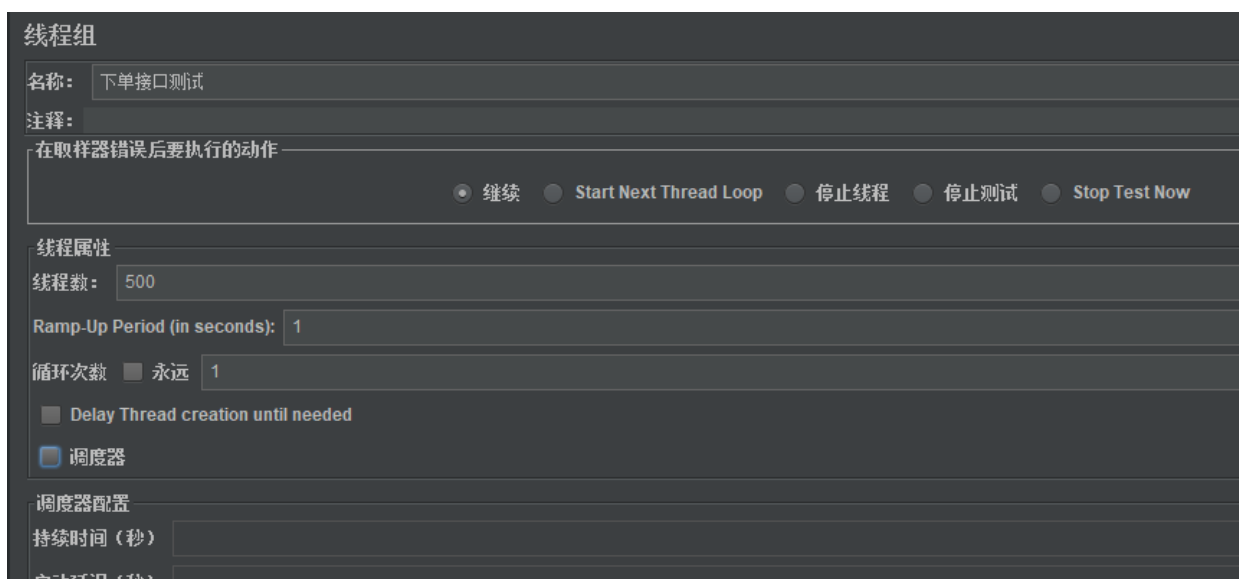
四.创建测试

1.创建线程组

在“测试计划”上右键【添加】-->【Threads(Users)】-->【线程组】。

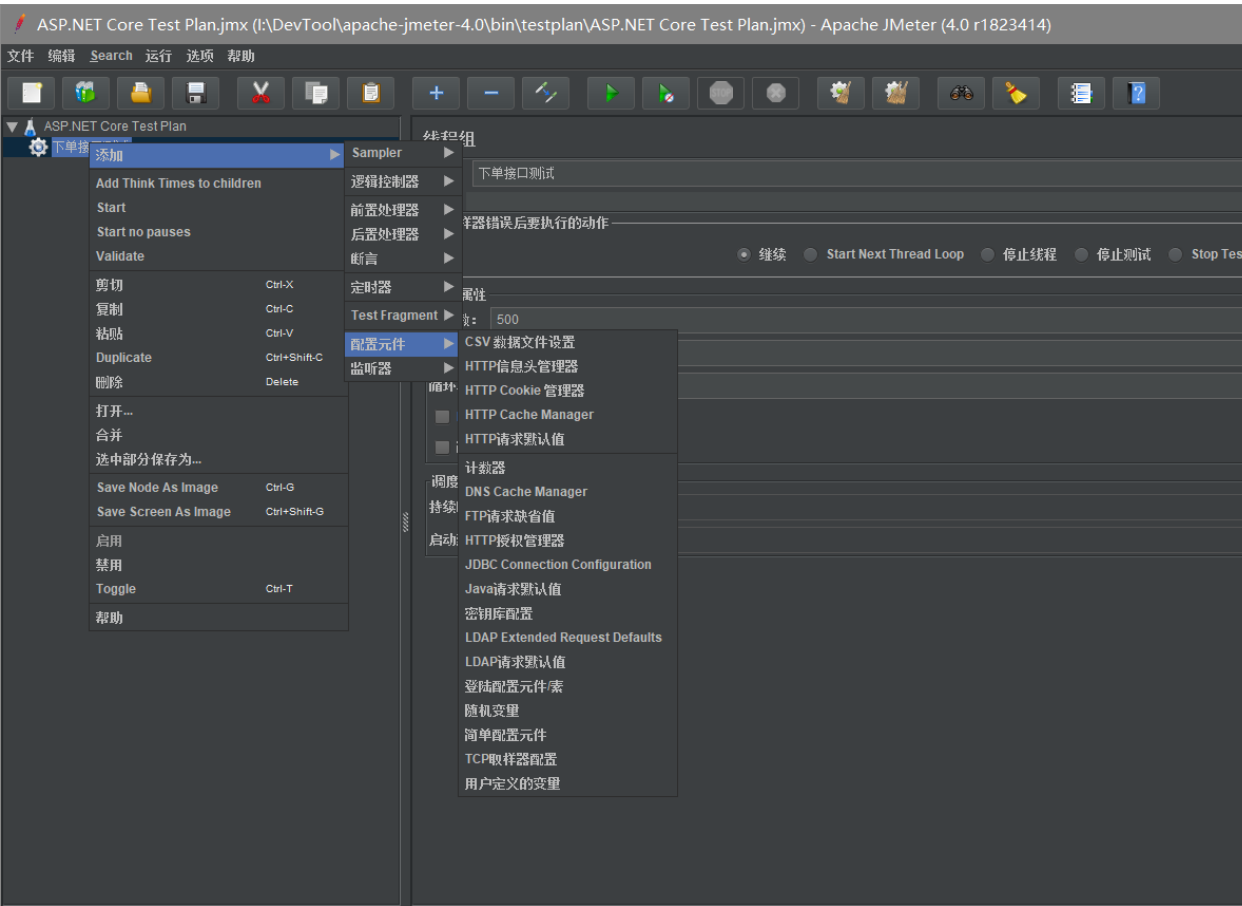


设置线程数和循环次数。我这里设置线程数为500，循环一次。

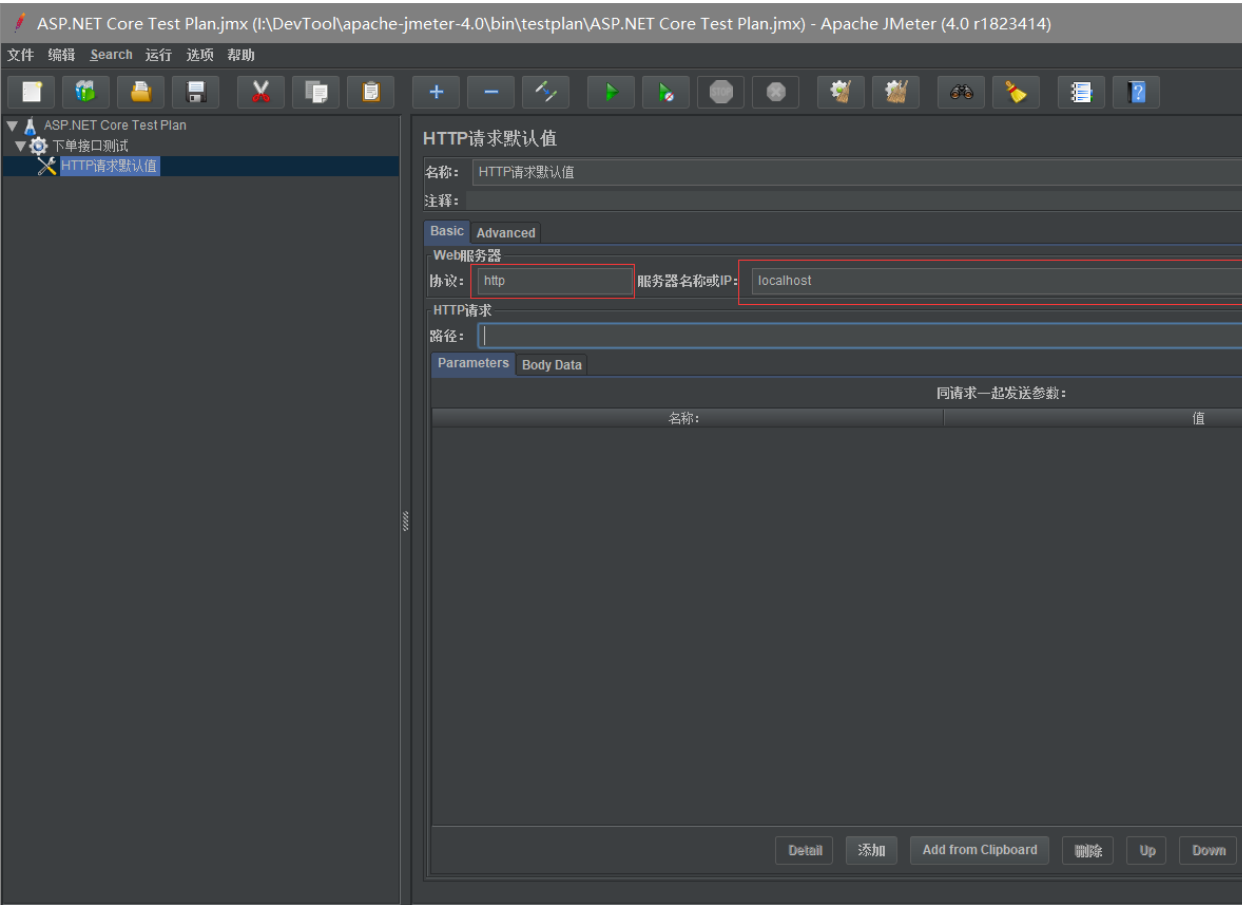


2. 配置元件

在我们刚刚创建的线程组上右键 【添加】-->【配置元件】-->【HTTP请求默认值】。



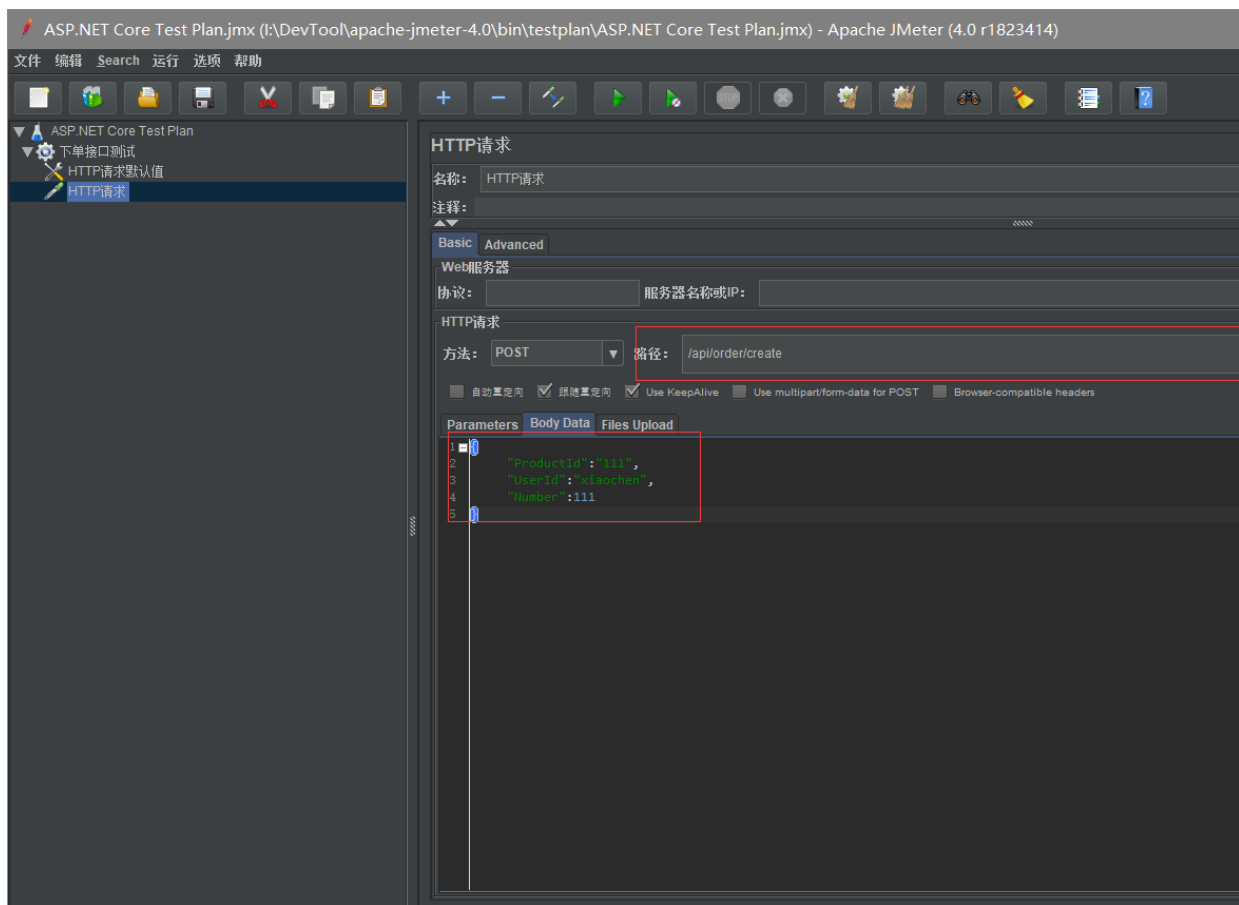
配置我们需要进行测试的程序协议、地址和端口



当所有的接口测试的访问域名和端口都一样时，可以使用该元件，一旦服务器地址变更，只需要修改请求默认值即可。

3.构造HTTP请求

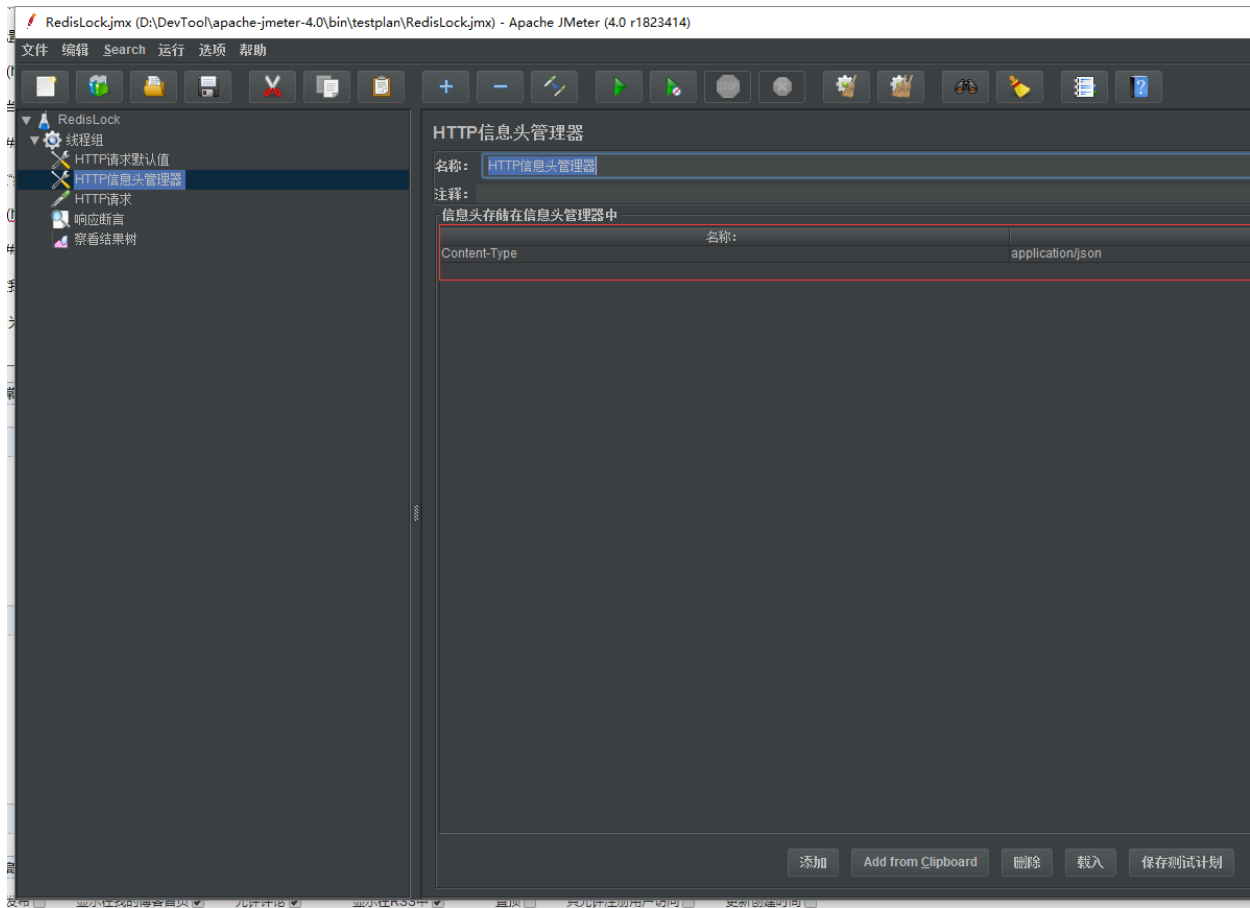
在“线程组”右键 【添加-】 -> 【sampler】 --> 【HTTP 请求】设置我们需要测试的API的请求路径和数据。我这里用的是json



4.添加HTTP请求头

在我们刚刚创建的线程组上右键 【添加】 --> 【配置元件】 --> 【HTTP信息头管理器】。

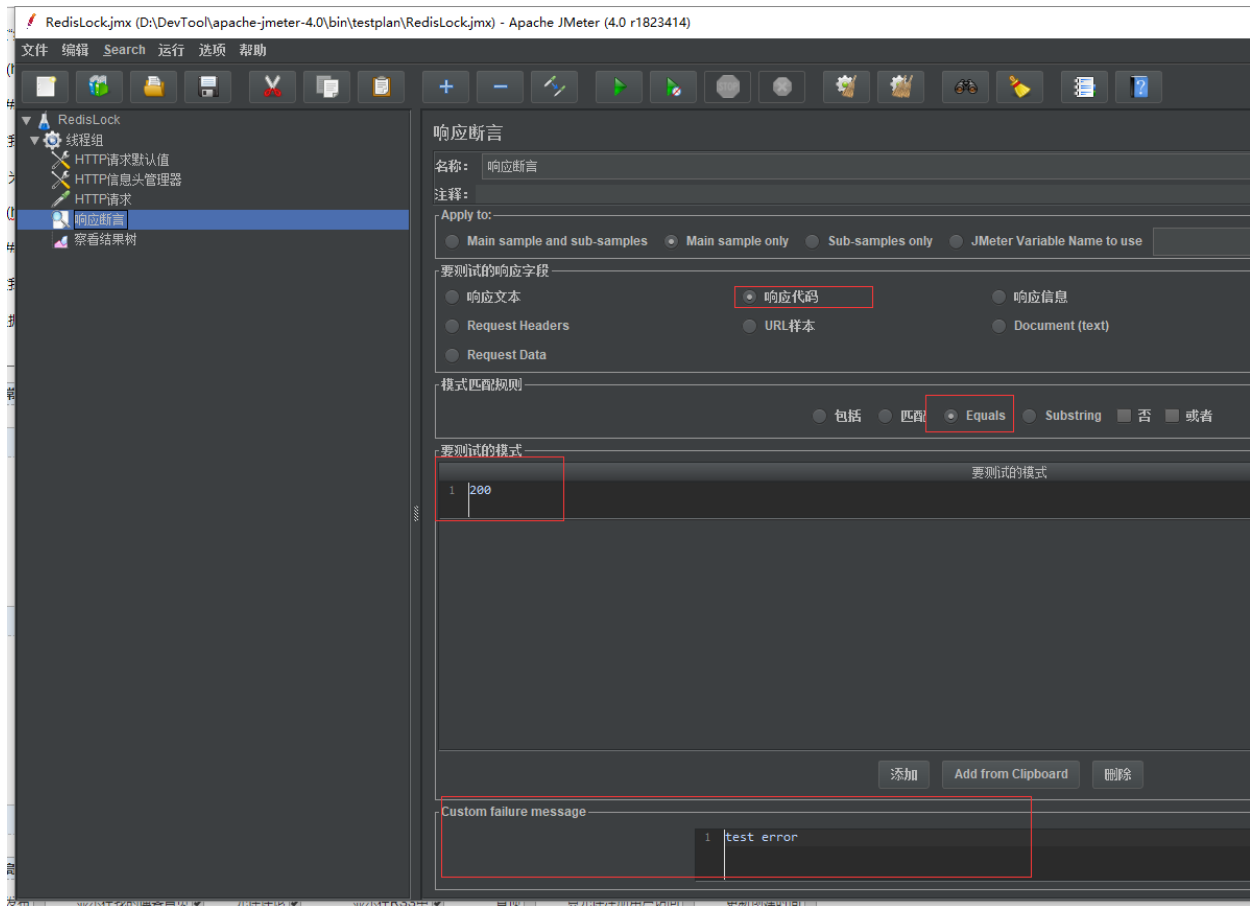
因为我要传输的数据为json，所以设置一个 `Content-Type:application/json`



5. 添加断言

在我们刚刚创建的线程组上右键【添加】-->【断言】-->【响应断言】。

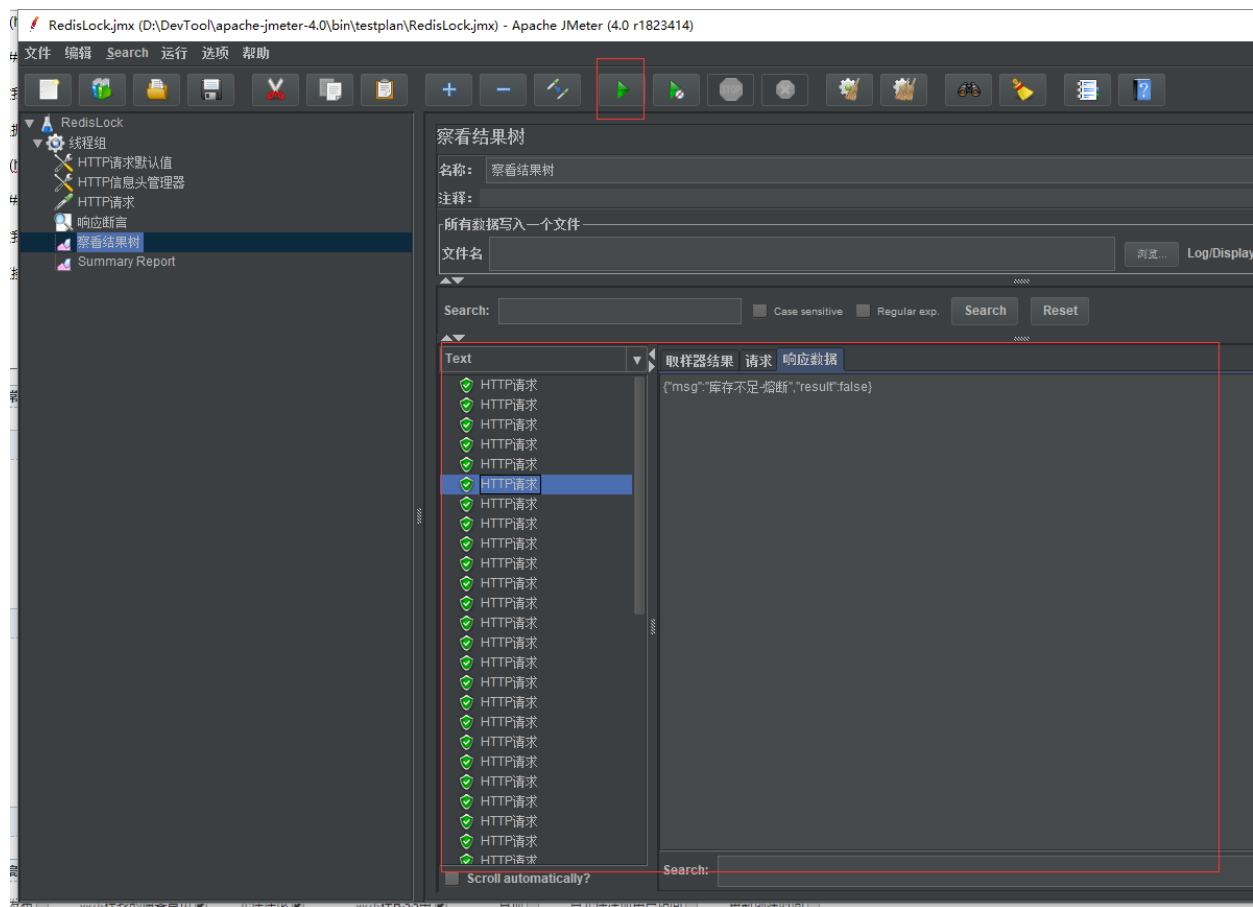
根据响应的数据来判断请求是否正常。我在这里只判断的响应代码是否为200。还可以配置错误信息



6. 添加察看结果树

在我们刚刚创建的线程组上右键【添加】-->【监听器】-->【察看结果树】。

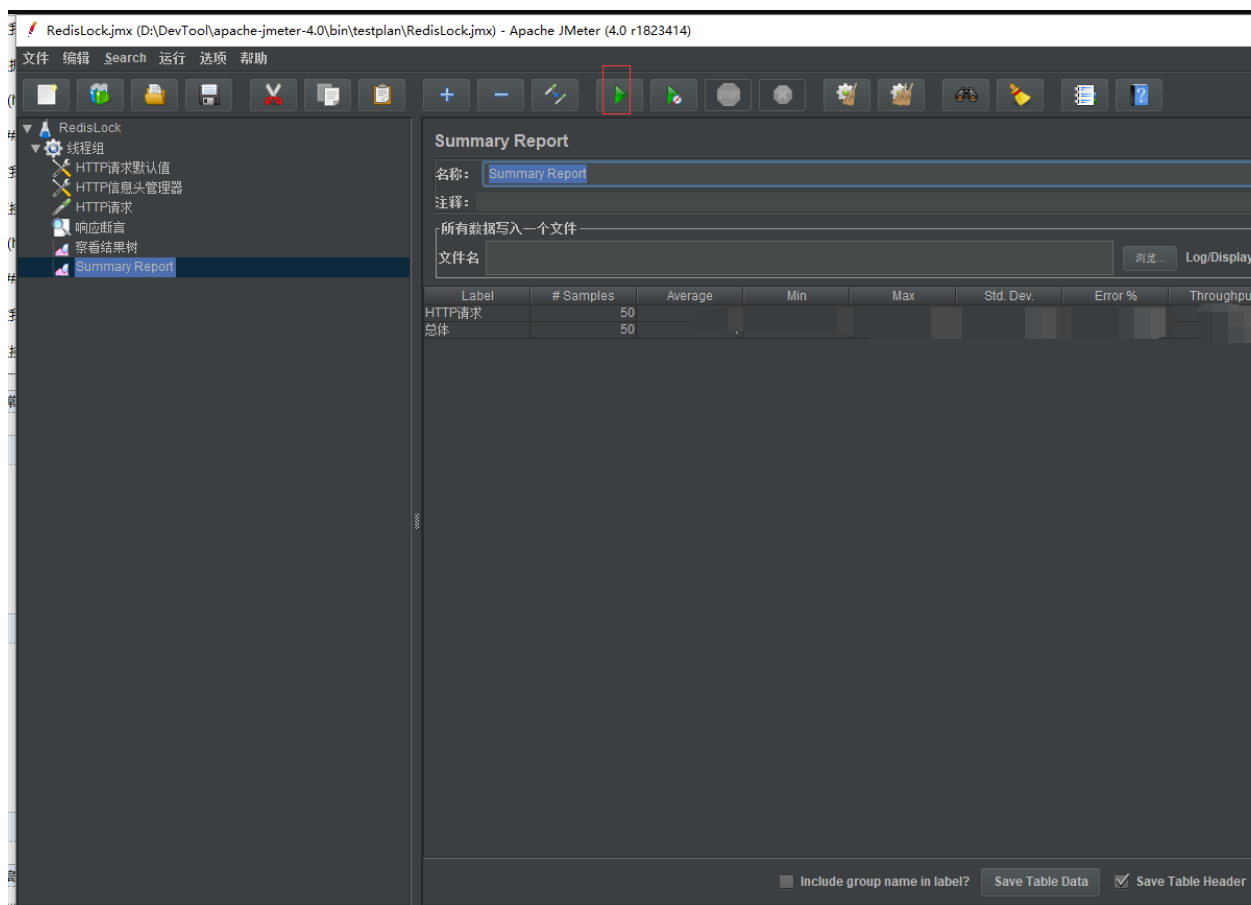
直接添加，然后点击 **运行** 按钮就可以看到结果了。



7. 添加Summary Report

在我们刚刚创建的线程组上右键【添加】-->【监听器】-->【Summary Report】。

直接添加，然后点击 **运行** 按钮就可以看到结果了。



为了不引起不必要的争论，隐藏了TPS。此数据不具备任何价值，仅仅为文章演示。

8.测试计划创建完成

记得点保存。

五.执行测试计划

前面我们说过，执行测试计划不能用GUI，需要用命令行来执行。

```
命令提示符
或批处理文件。

D:\DevTool\apache-jmeter-4.0\bin>jmeter -n -t testplan/RedisLock.jmx -l testplan/result/result.
ort
May 05, 2018 2:26:30 PM java.util.prefs.WindowsPreferences <init>
WARNING: Could not open/create prefs root node Software\JavaSoft\Prefs at root 0x80000002. Wind
returned error code 5.
Creating summariser <summary>
Created the tree successfully using testplan/RedisLock.jmx
Starting the test @ Sat May 05 14:26:30 CST 2018 (1525501590435)
Waiting for possible Shutdown/StopTestNow/Heapdump message on port 4445
summary +      1 in 00:00:00 =      0/s Avg:   57 Min:   57 Max:   57 Err:    0 (0.00%) Acti
d: 0
summary +     49 in 00:00:01 =     49.7/s Avg:    6 Min:    1 Max:   57 Err:    0 (0.00%) Acti
ed: 50
summary =     50 in 00:00:01 =     50.8/s Avg:    7 Min:    1 Max:   57 Err:    0 (0.00%)
Tidying up ... @ Sat May 05 14:26:31 CST 2018 (1525501591782)
... end of run
D:\DevTool\apache-jmeter-4.0\bin>
```

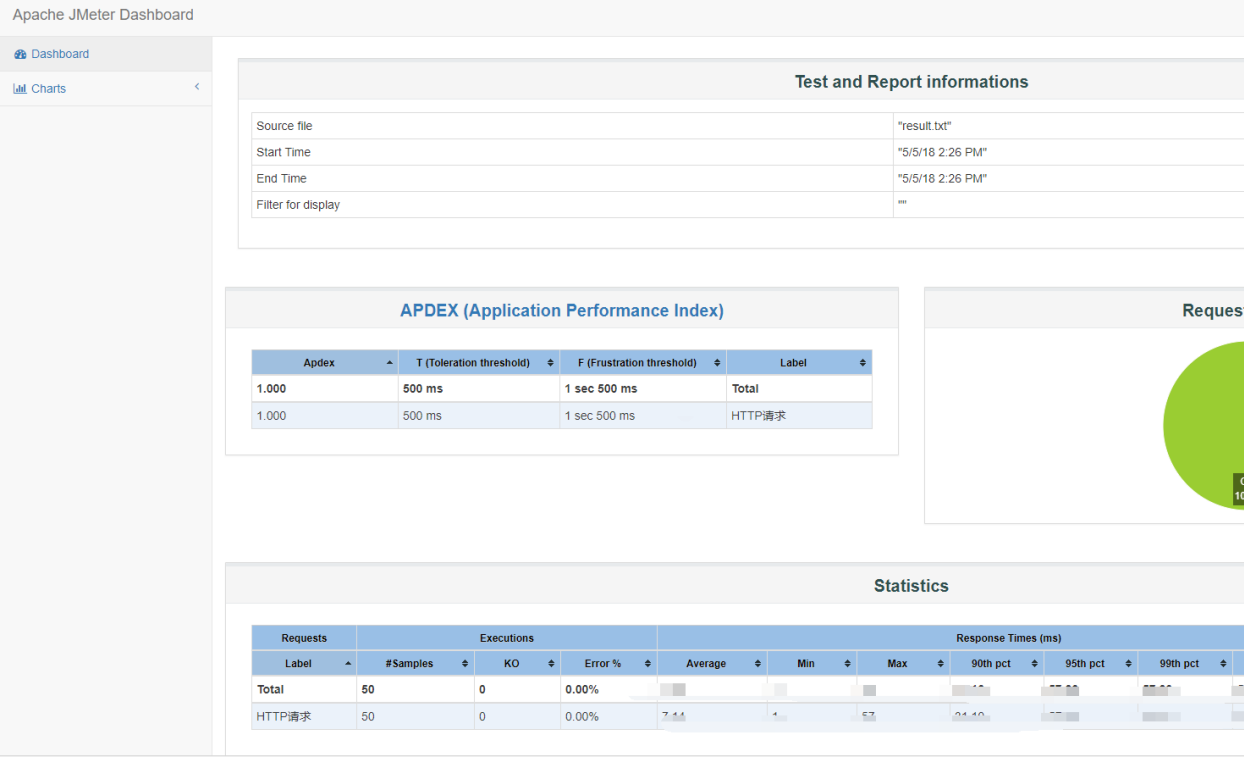
我这里执行的命令为：

```
jmeter -n -t testplan/RedisLock.jmx -l testplan/result/result.txt -e -o testpla
```

说明：

- testplan/RedisLock.jmx 为测试计划文件路径
- testplan/result/result.txt 为测试结果文件路径
- testplan/webreport 为web报告保存路径。

Web报告如下：



六.写在最后

线程数量和循环次数将会影响最终的测试报告,请大家多多测试。