

信息科学与工程学院

软件工程 实验报告

**系 别 信息科学与工程学院**

**专 业 计算机科学与技术**

**年 级 2021级**

**姓 名 徐昊博**

**指导教师 阮彤**

**2024 学年 第 二 学期**

**实验九、性能测试**

1. **实验目的**

（1）了解Jmeter性能测试工具的用途和简单的操作；

（2）掌握Jmeter性能测试工具测试过程；

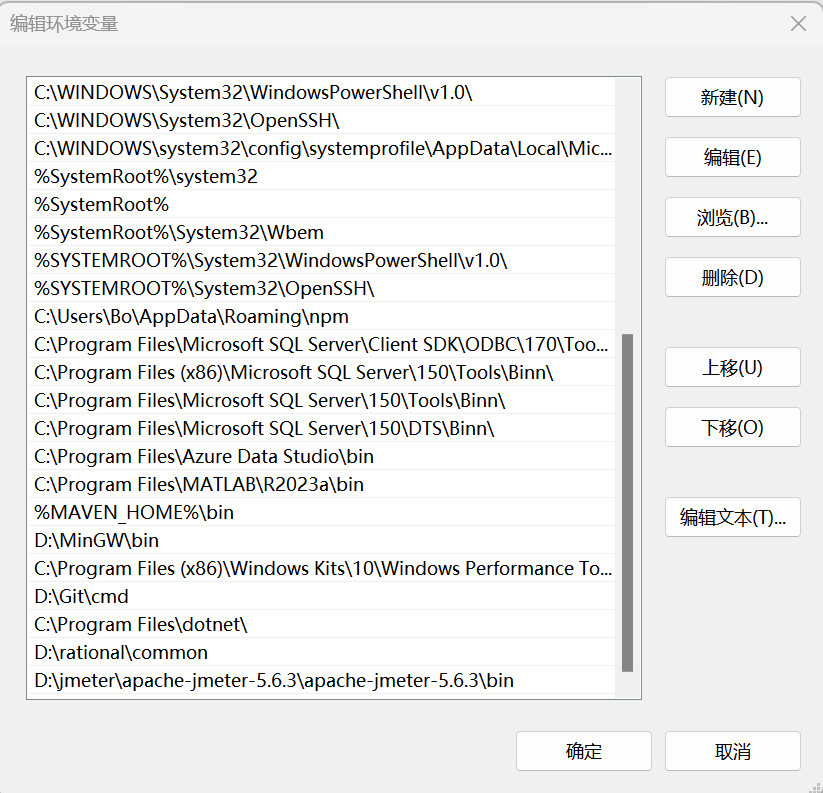
（3）能够使用Jmeter进行负载测试；

**二、实验主要内容**

下载并安装Jmeter，在架构Demo系统上进行测试脚本的编写，进行性能测试，并对测试结果进行分析

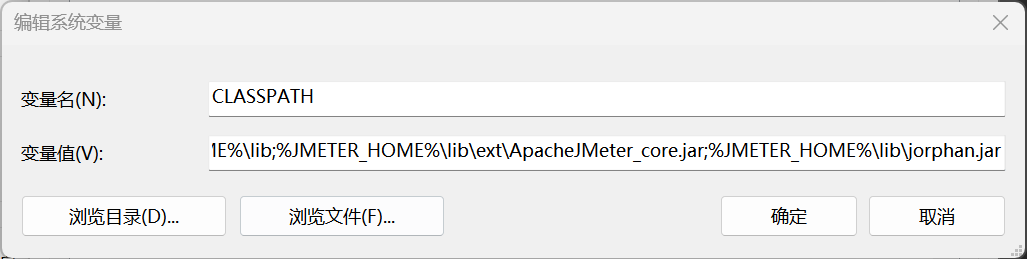
**三、实验过程**

1、前往官网进行下载<https://jmeter.apache.org/download_jmeter.cgi>，并解压压缩包到英文目录下并配置环境变量：（1）设置jmeter解压目录的JMETER\_HOME环境变量系统变量中新建变量，变量名为JMETER\_HOME，变量值为jmeter安装的路径(2) 设置jmeter中bin目录的path环境变量在系统的PATH变量中添加jmeter安装目录下bin目录的路径

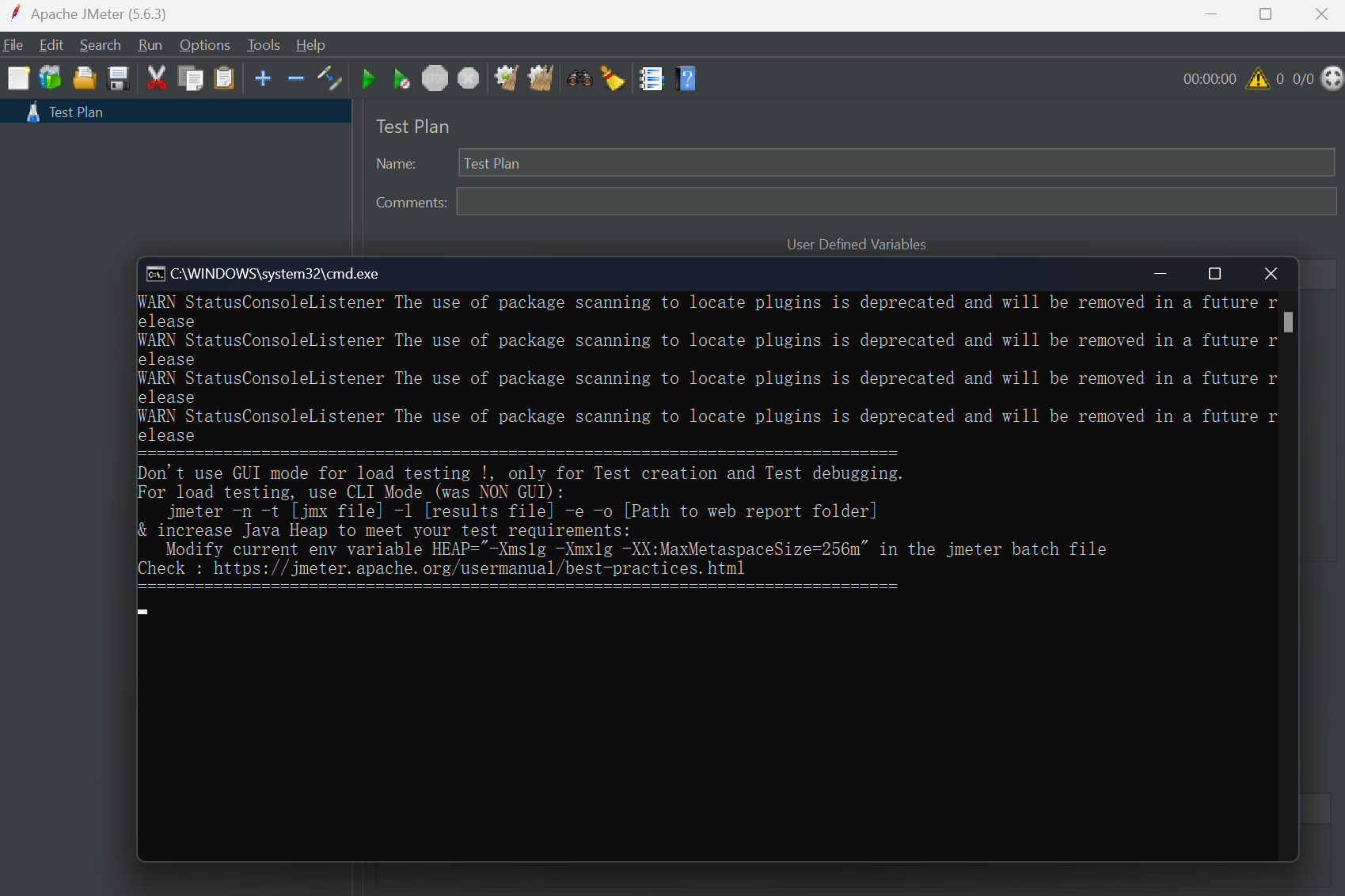
****

（3）设置jmeter的classpath变量

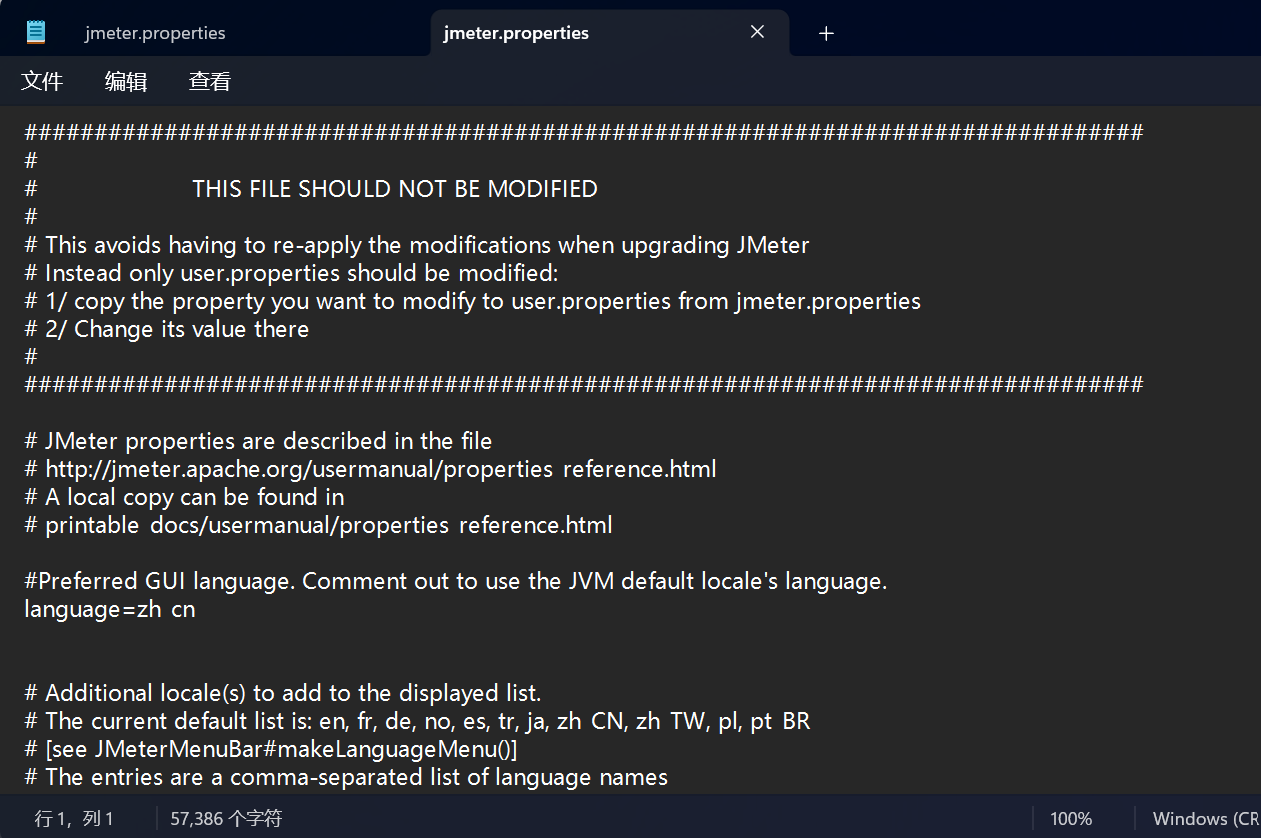
系统属性中找到一个变量名为CLASSPATH的系统变量（若没有则新建），新增两条：%JMETER\_HOME%\lib\ext\ApacheJMeter\_core.jar;%JMETER\_HOME%\lib\jorphan.jar;

****

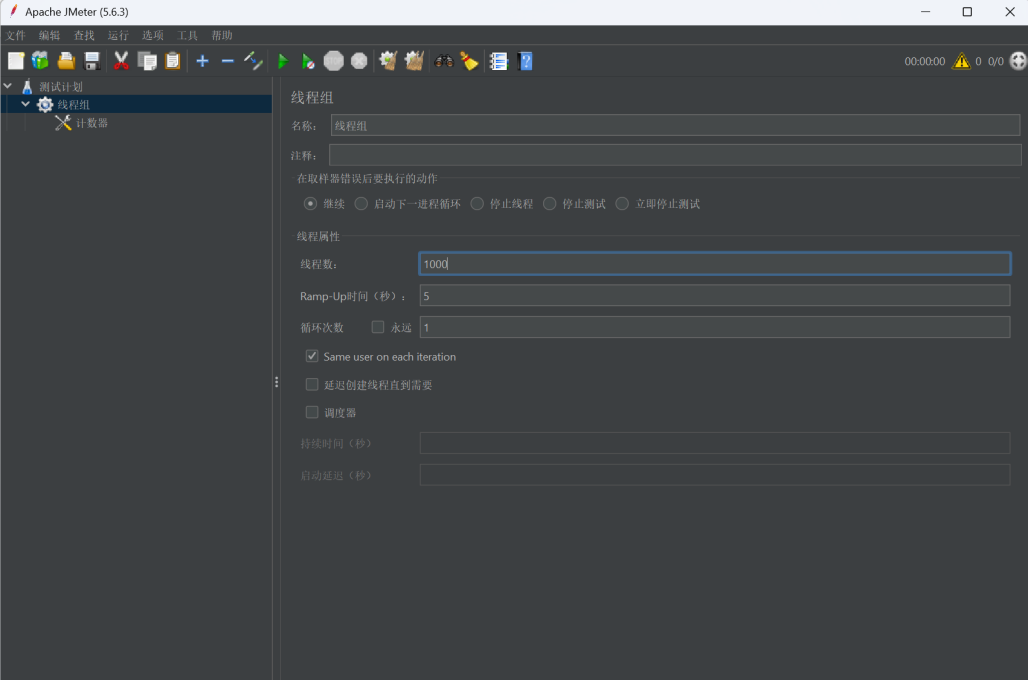
（4）进入jmeter下的bin目录，点击打开jmeter.bat文件，即可启动jmeter。

****

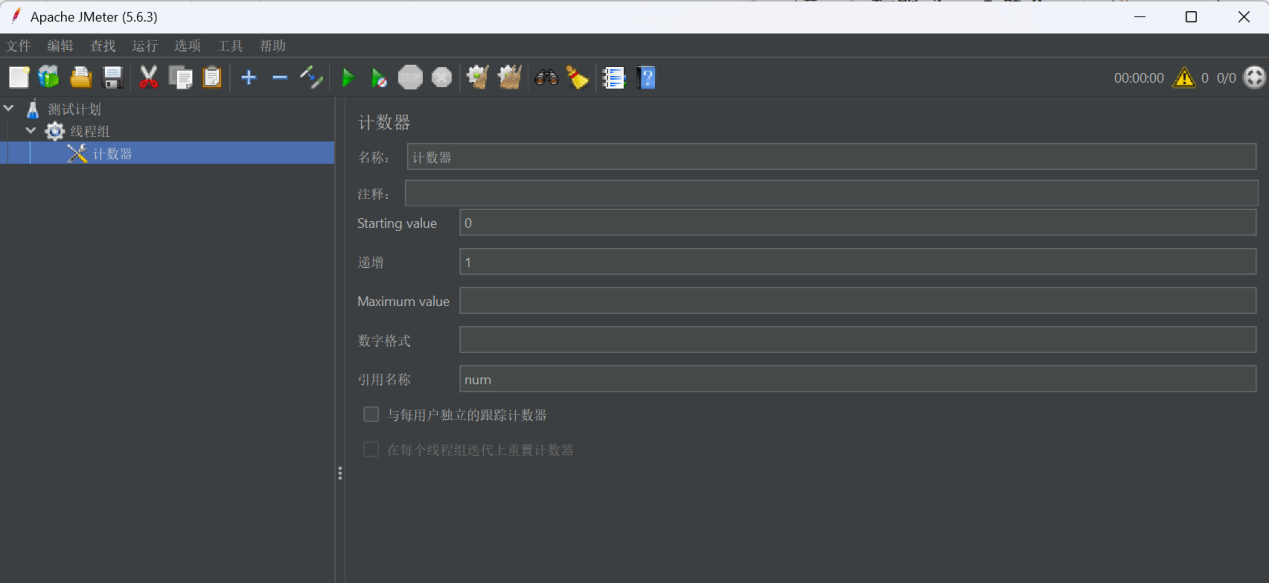
（5）修改配置文件 jmeter.properties（通过记事本、EditPlus等文件打开），添加或修改以下字段：**language = zh\_cn**，随后重新启动即可完成汉化。

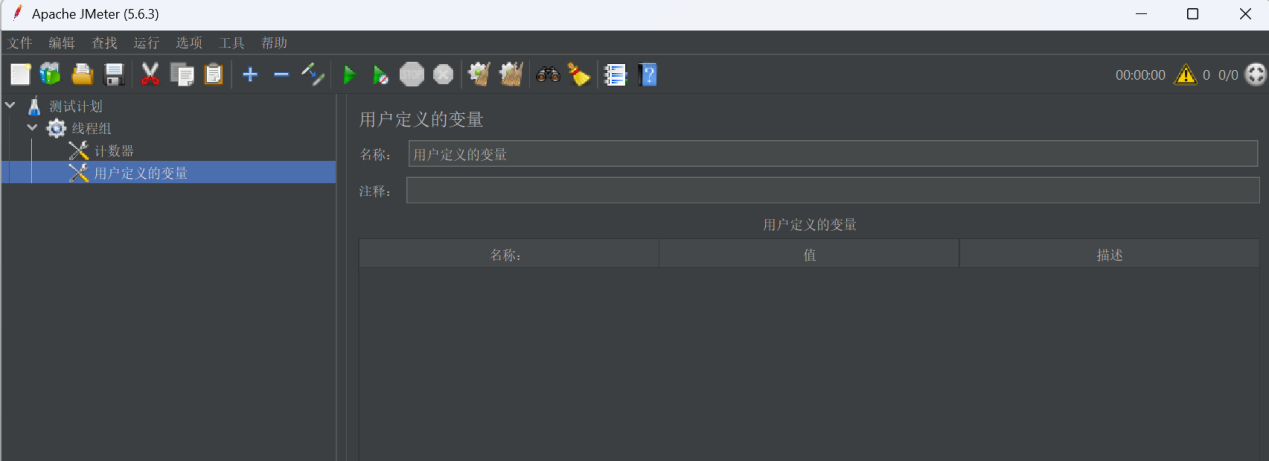
****

2、开始测试  
（1）在测试计划中添加取样器中的”线程组”，设置线程数为1000，Ramp-Up时间为5秒

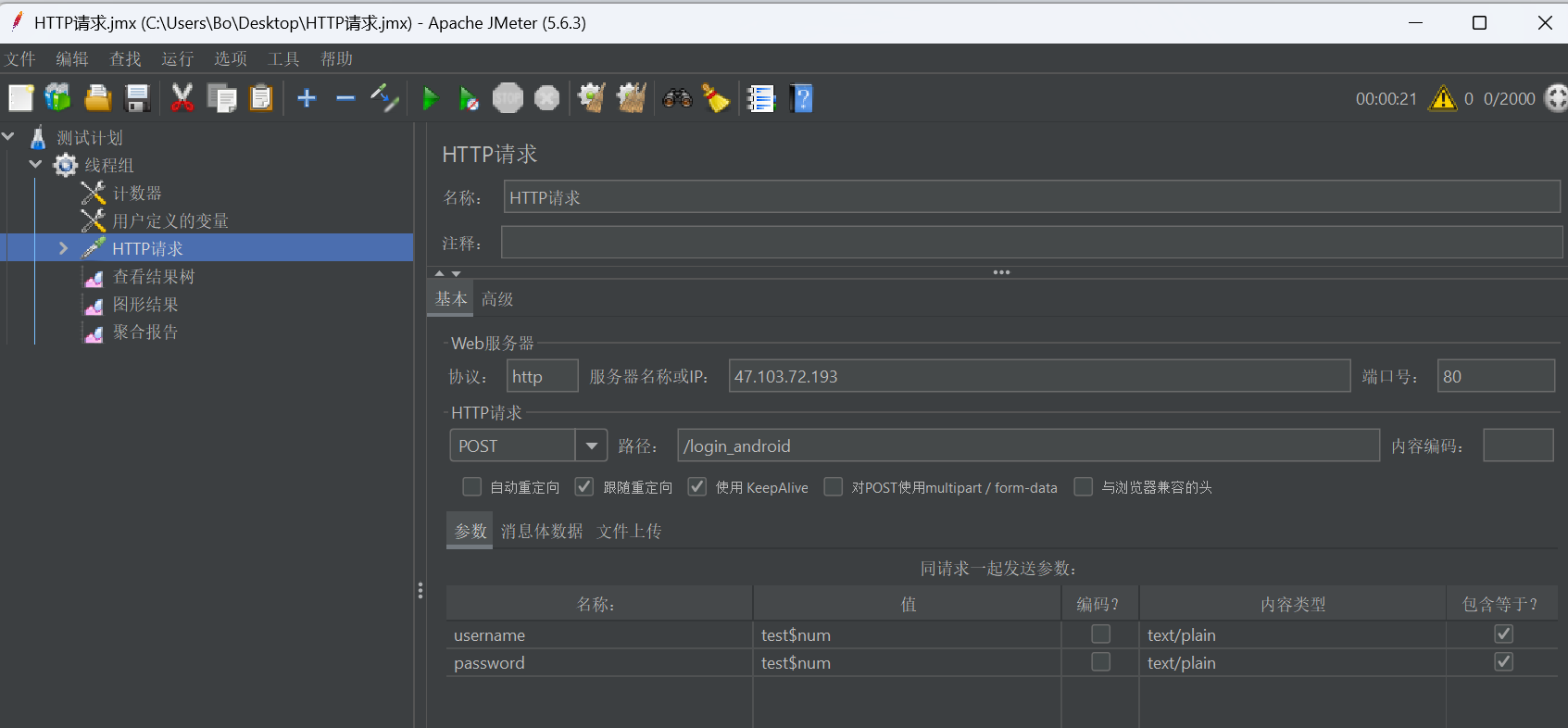
****

（2）为了保证数据库主键的唯一性，并使生成的数据有规律，便于查看，在创建的线程组下添加配置元件中的”计数器”，初始值设置为0，递增为1，并设置引用名称为num，就可以在其他元件中通过${num}获取计数器的值

****

（3）用户可以在线程组下添加配置元件中的”用户已定义的变量”，对需要测试的数据进行封装。****

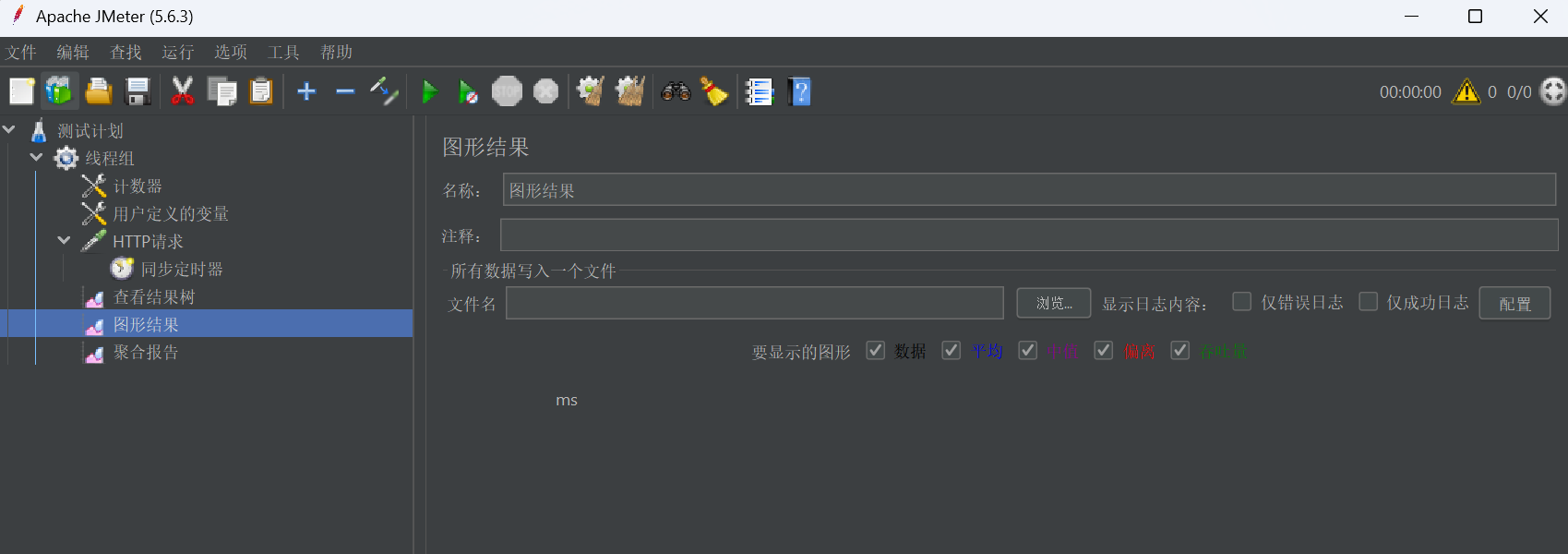
（4）在线程组下添加取样器中的HTTP请求，根据上述信息填写web服务器、HTTP请求以及发送的参数名称和值，如果需要调用测试计划中定义的变量，需要以${名称}格式调用(如${num}，${id})。



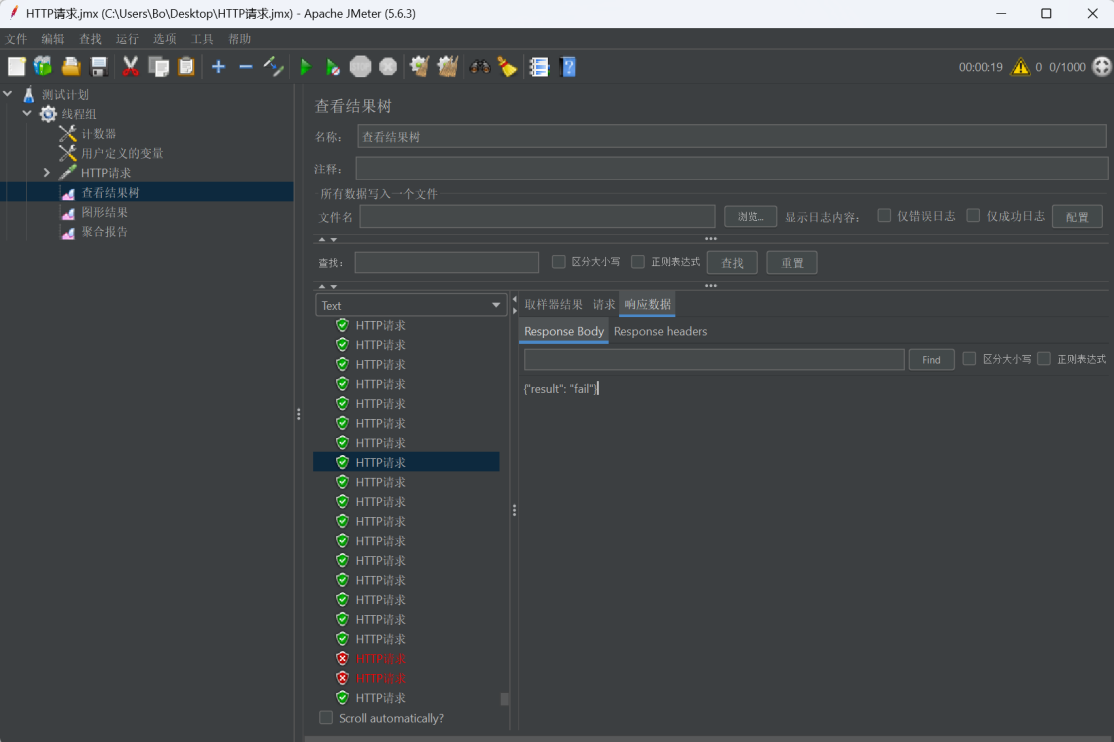
（5）在HTTP请求下添加定时器中的”同步定时器（synchronized timer）”，设置模拟用户组的数量为500，使500个线程都生成后统一发送请求

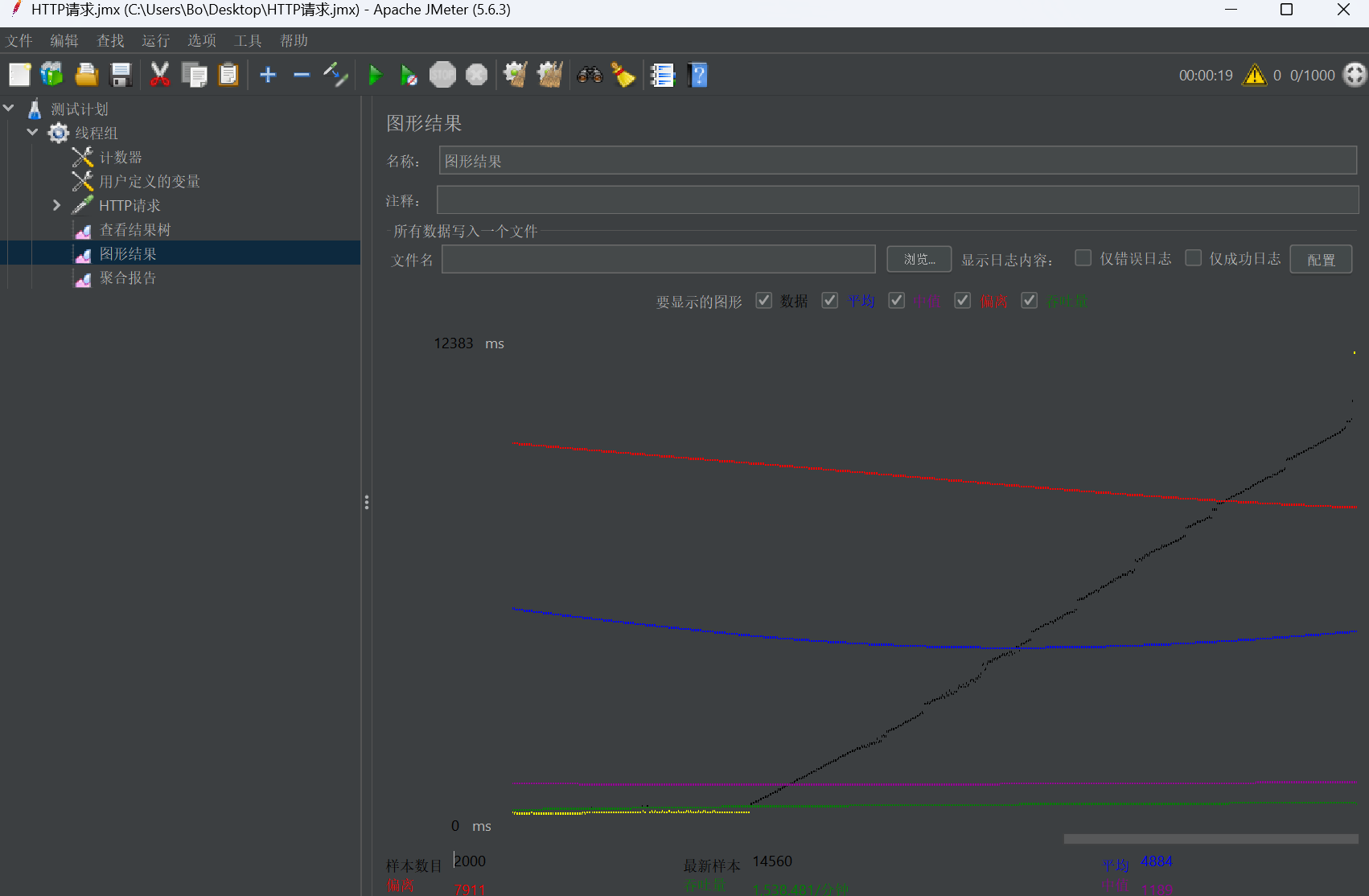
****

（6）在线程组下分别添加监听器中的察看结果树、图形结果以及聚合报告，在聚合报告中填写一个文件名，使测试结果输出到文件中

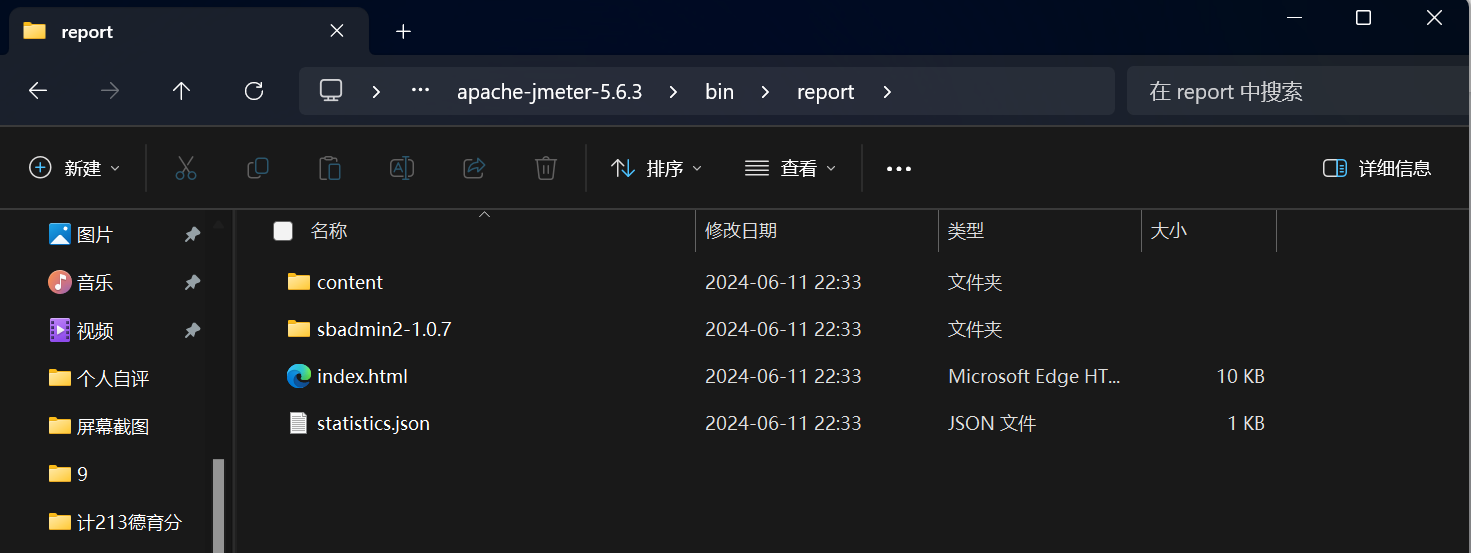
****



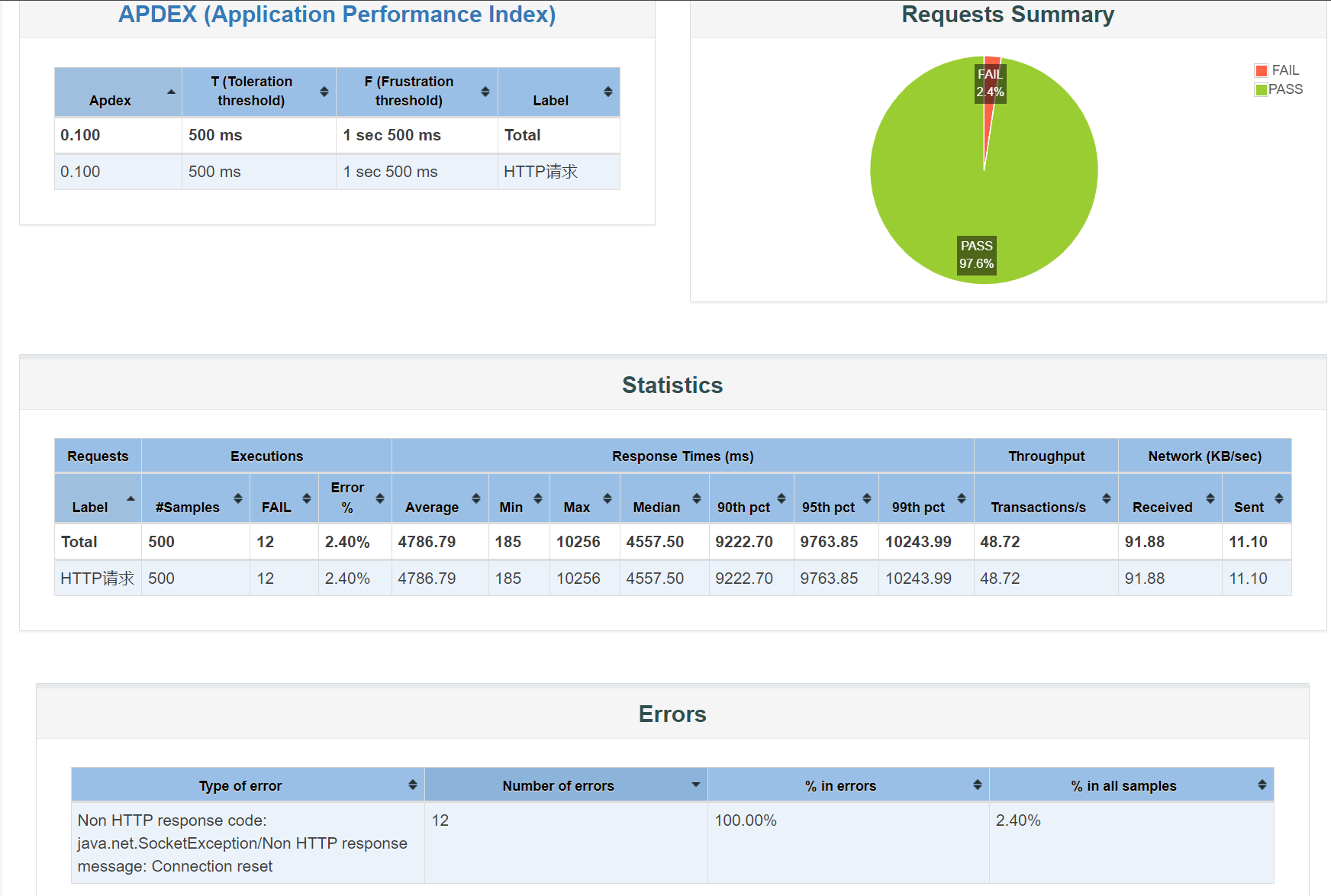
（7）点击播放键开始测试，在网页、察看结果树、图形结果、聚合报告中查看测试结果：****

****

（8）将生成的jtl测试结果文件转换成html，使用命令行前往jmeter安装路径下的bin文件夹中，执行jmeter -g 生成的jtl文件 -o 输出路径（输出路径为一个空的文件夹）

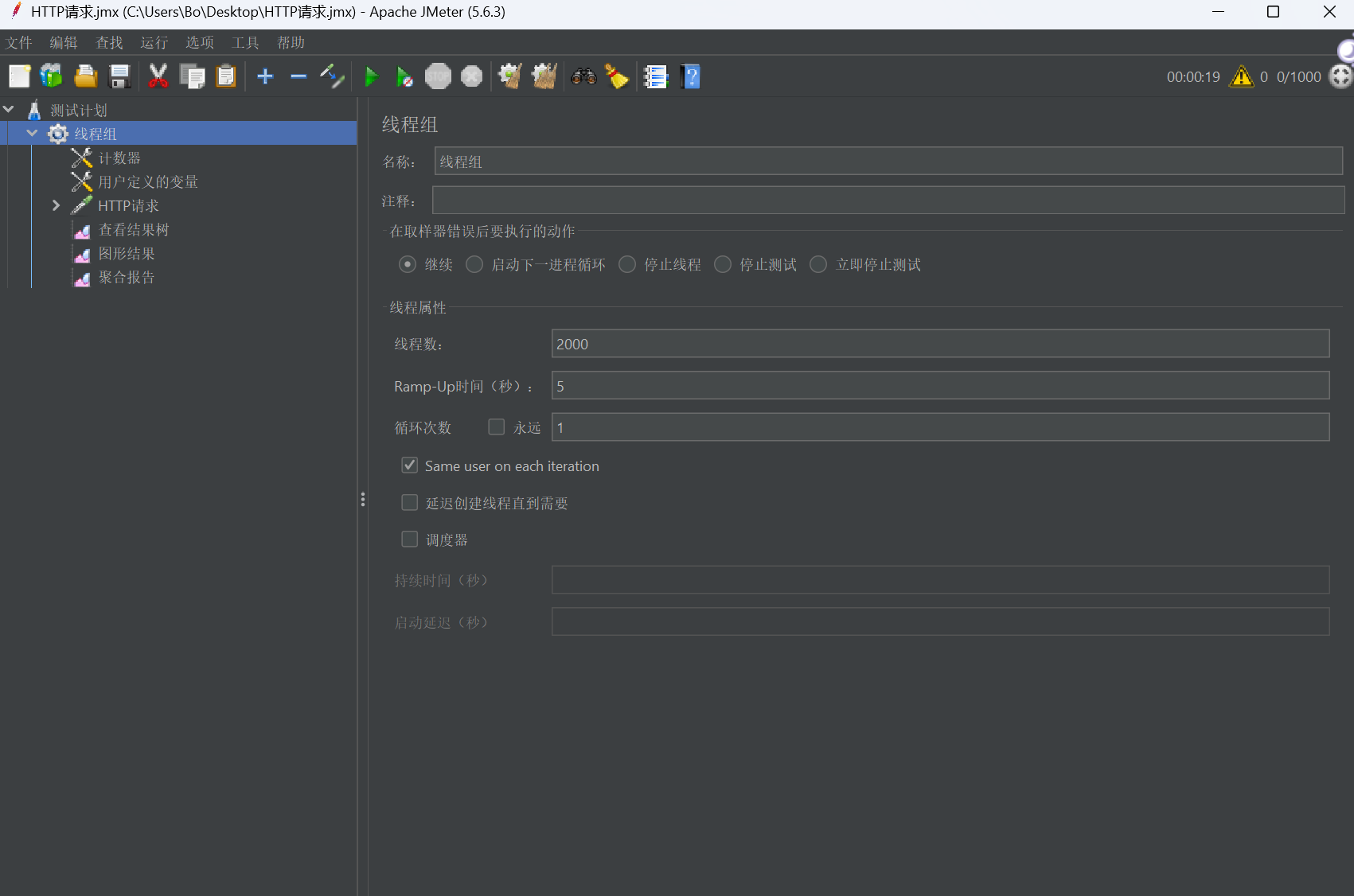


（9）执行命令后，即可在输出路径看到结果，打开index.html即可看到全面的测试报告



1. 异常测试情况

（1）修改线程组的线程数为2000，同步定时器模拟用户组的数量为2000

****

（2）执行测试可以看到异常率相较之前的2.4%提高到了70.9%可见服务器的容量已经出现负载过高的情况

****

**四、实验总结和收获**

实验总结

本次实验的主要目的是通过使用Jmeter工具进行性能测试，了解其用途、操作过程，并掌握如何进行负载测试。在实验过程中，我们经历了Jmeter的安装和配置，创建测试计划，添加HTTP请求和监听器，执行测试计划，最后分析测试结果的完整流程。

1. Jmeter的用途和操作：

通过实验，我们初步了解了Jmeter作为一个开源的性能测试工具，能够用于不同类型的测试，如负载测试、功能测试、回归测试等。其界面直观，操作简便，使得新手也能够快速上手。

2. 测试过程的掌握：

通过详细的实验步骤，我们掌握了如何在Jmeter中创建一个测试计划，包括添加线程组、配置HTTP请求以及添加监听器等。特别是对线程组的设置和HTTP请求的配置有了更深入的理解。

3. 负载测试的实践：

- 在进行负载测试时，我们学会了如何调整线程组中的线程数和循环次数，以模拟不同的负载条件。通过观察不同负载条件下的系统响应时间、吞吐量和错误率等指标，我们能够更直观地评估系统的性能表现。

实验收获

1. 实用的工具知识：

本次实验使我们掌握了一个重要的性能测试工具——Jmeter的基本使用方法。这为我们在实际工作中进行性能测试打下了坚实的基础。

2. 性能测试的重要性：

实验过程中，通过模拟不同负载条件下的系统运行情况，我们认识到性能测试在软件开发和运维中的重要性。它不仅能帮助我们发现系统的性能瓶颈，还能为系统优化提供有价值的数据支持。

3. 问题发现与解决：

在实验中，我们遇到了一些实际问题，例如HTTP请求的配置错误、监听器数据显示不正确等。通过查找资料和实际操作，我们学会了如何解决这些问题，提高了我们的问题解决能力。

4. 数据分析能力：

通过分析测试结果数据，我们增强了对性能指标（如响应时间、吞吐量、错误率等）的理解。学会了如何从数据中发现问题，并针对性地进行优化。

实验心得

通过本次实验，我们不仅学会了如何使用Jmeter进行性能测试，还深刻体会到了性能测试在实际项目中的应用和价值。未来，在进行系统开发和运维时，我们会更加注重性能测试，确保系统能够在各种负载条件下稳定运行。同时，我们也将继续学习和深入探索Jmeter的高级功能，提升我们的测试技能和数据分析能力。