JMeter的简单使用说明

# JMeter简介

Apache JMeter是Apache组织开发的基于Java的测试工具。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。

# 安装与配置

## 工具/原料

* JDK 1.7
* Apache JMeter 3.0

## 环境

JMeter可以在Windows/Linux/Mac平台上运行，这里介绍它的安装配置及测试。

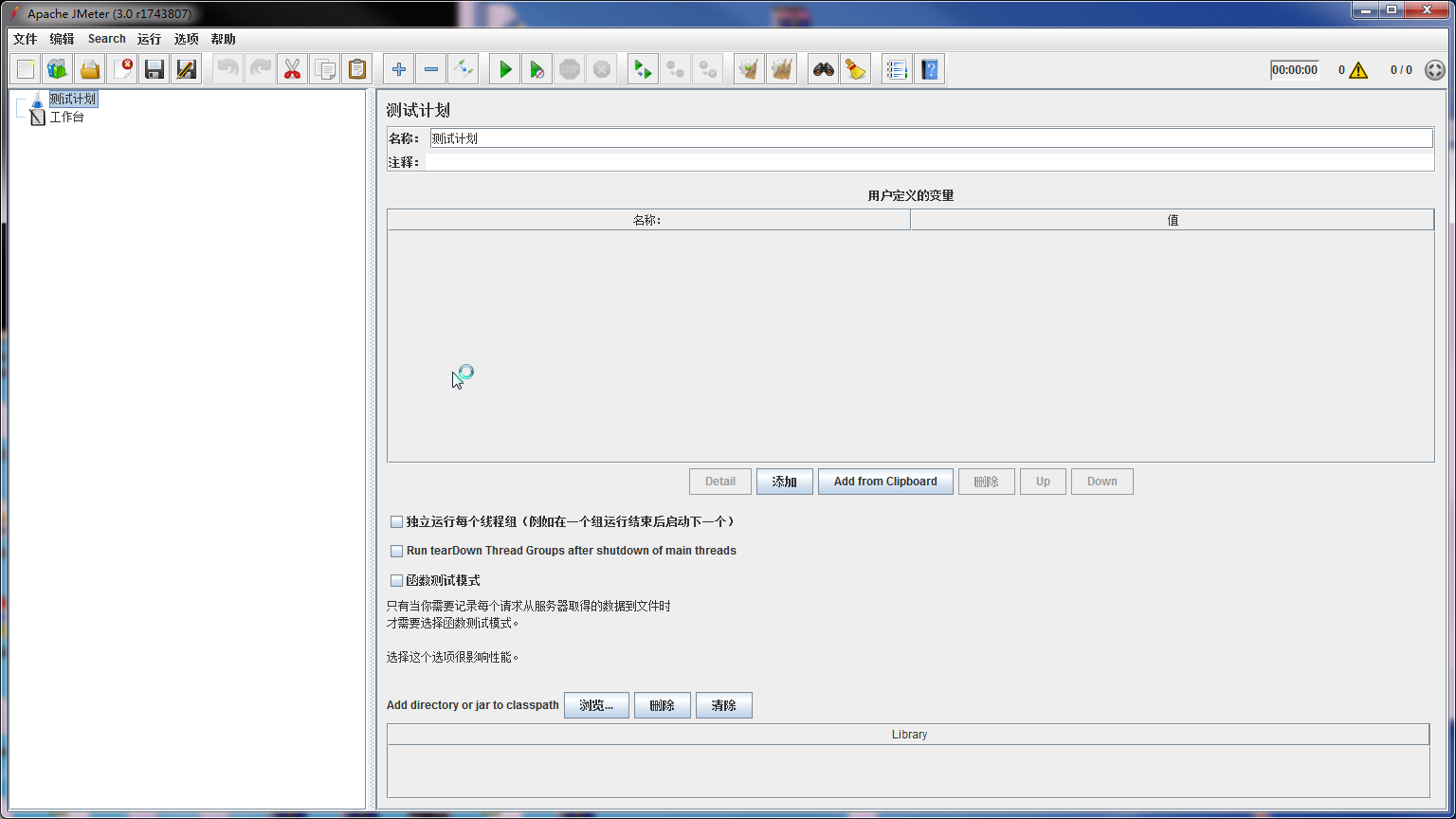
## 安装步骤

* 安装JDK，网上下载一个JDK，配置好环境变量即可使用。在dos界面输入java -version，能显示版本号，则证明JDK安装OK，可以使用。



* 安装Jmeter：Jmeter不需要安装，解压之后即可使用；

双击Jmeter的bin目录下的jmeter.bat，出现下面界面，则Jmeter启动成功，Jmeter就绪。



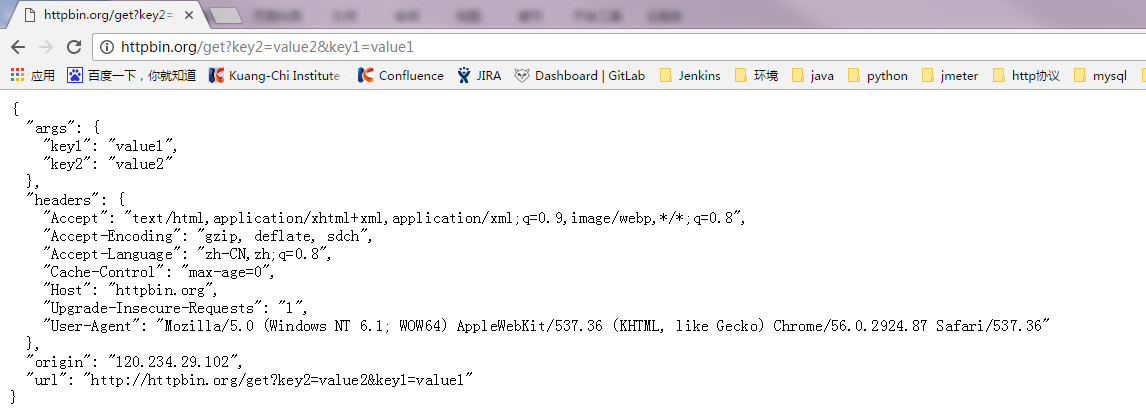
# HTTP接口测试

## 测试实例

以一个简单的接口为例：

<http://httpbin.org/get?key2=value2&key1=value1>

在浏览器中打开的效果如下：

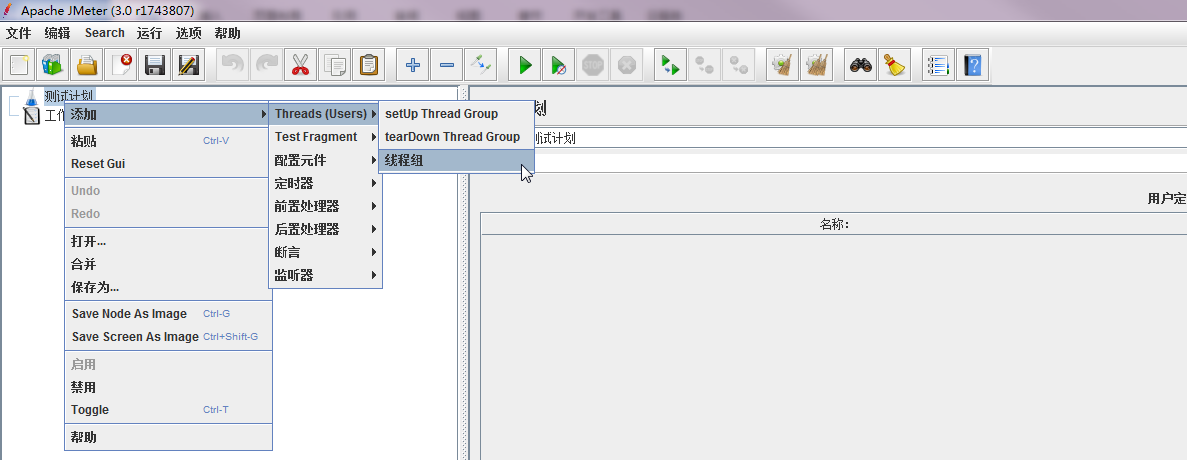


在这里，成功的用浏览器访问了这个接口，并得到了成功的返回报文。下面我们用Jmeter来访问这个接口。

## 使用Jmeter进程HTTP测试

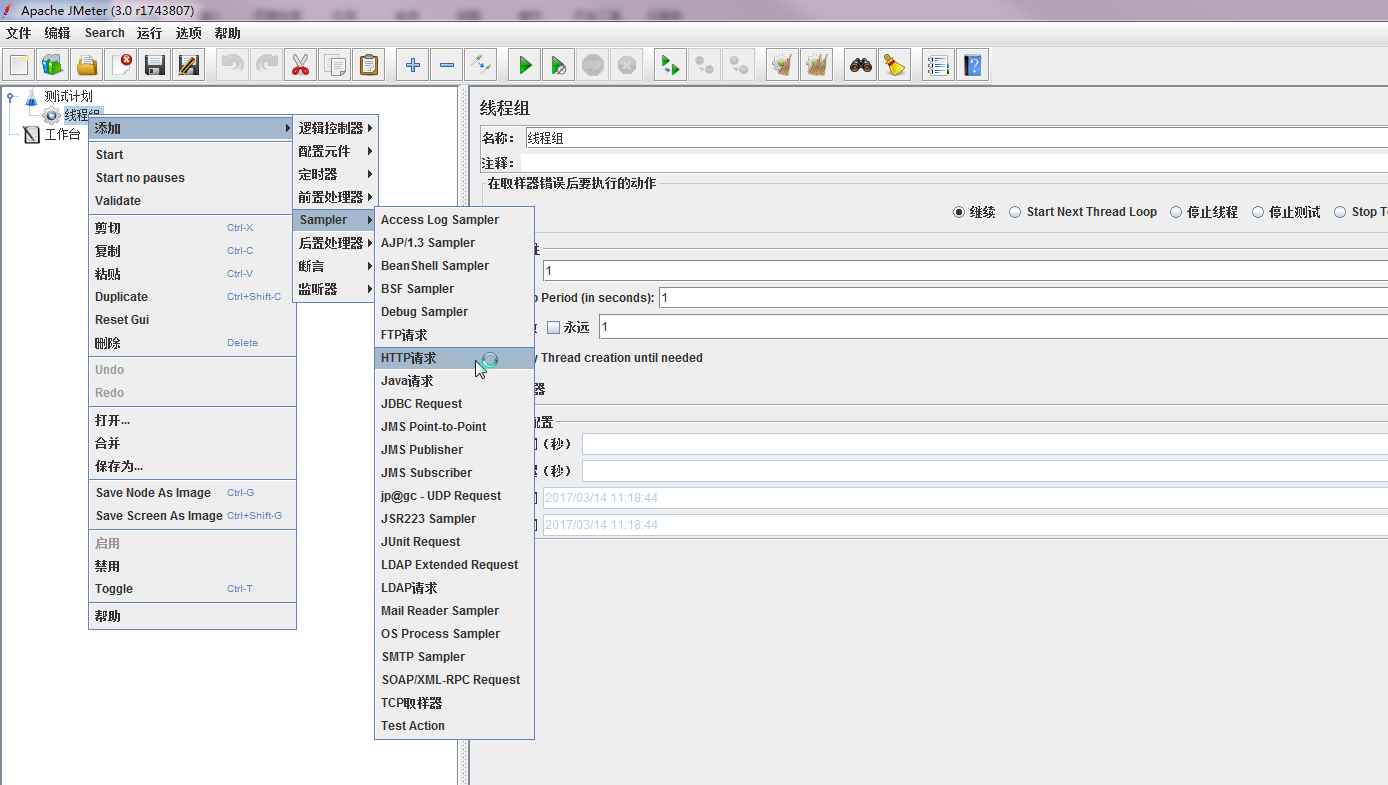
### 添加线程组

* 右键“测试计划”->“添加”->“Threads(Users)”->“线程组”

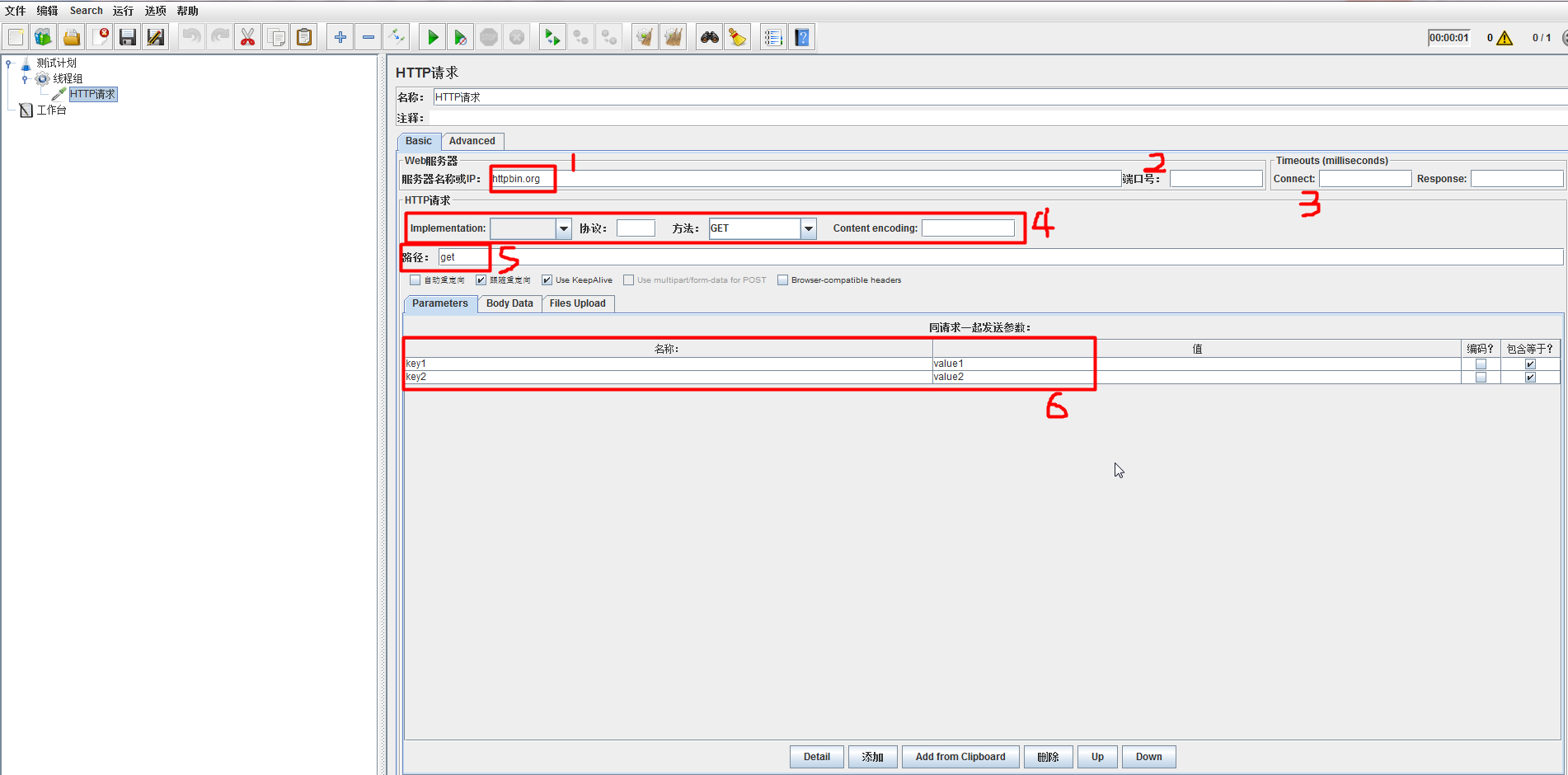


### 添加“HTTP请求”

* 右键“线程组”->“添加”->“Sampler”->“HTTP请求”



* 打开之后，填入访问该接口所需要的参数，效果如下图



Tips：

1. 服务器名或IP：这里填写服务IP地址（不带端口），有域名的直接填写域名
2. 端口号：这里填写端口号，有域名的这里为空就好。
3. Timeouts(milliseconds)：超时时间，单位是毫秒；Connect：连接超时；Response：响应超时。二者都可以不填
4. Implementa：采用什么实例来运行这次HTTP请求（HttpClient4/HttpClient3.1/Java），这里我们不用选择，直接默认就好。

协议：该接口采用什么协议（http/https），这里不填默认是http，根据情况填写

方法：请求该接口的方法（GET/POST等），根据接口请求类型选择。

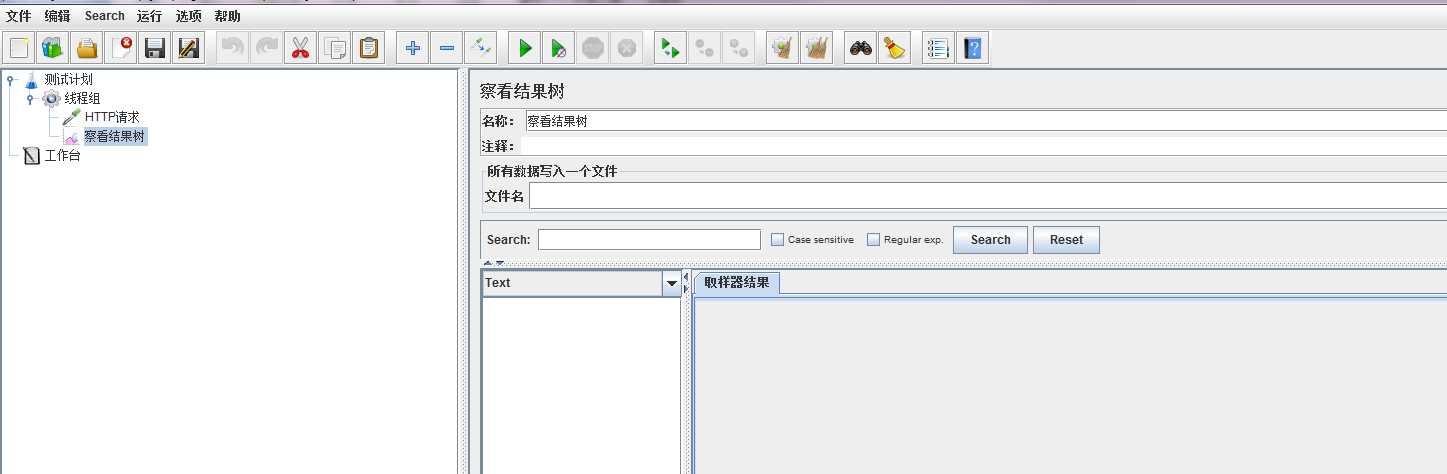
Content encoding：这里是返回报文的编码格式（如‘utf-8’），采用默认值就好

1. 同请求一起发送参数：接口的参数，可以通过下方“添加”按钮，添加参数，左边为“key”，右边填“value”，如上图。（这里只适用于传参数的接口，传Json和文件等接口使用这种方式不支持，暂不做介绍）

### 添加监听器

完成上述3.2.1和3.2.2步，一个接口的简单脚本已经编写完成了；但是此时我们还不知道在哪里查看脚本运行的结果，这里我们需要添加一个监听器。

* 右键“线程组”->“添加”->“监听器”，选择“察看结果树”



Tips：Jmeter提供了需要监听器，有兴趣的可以去一一添加，察看效果。

### 运行脚本

* 点击按钮，运行脚本，察看结果



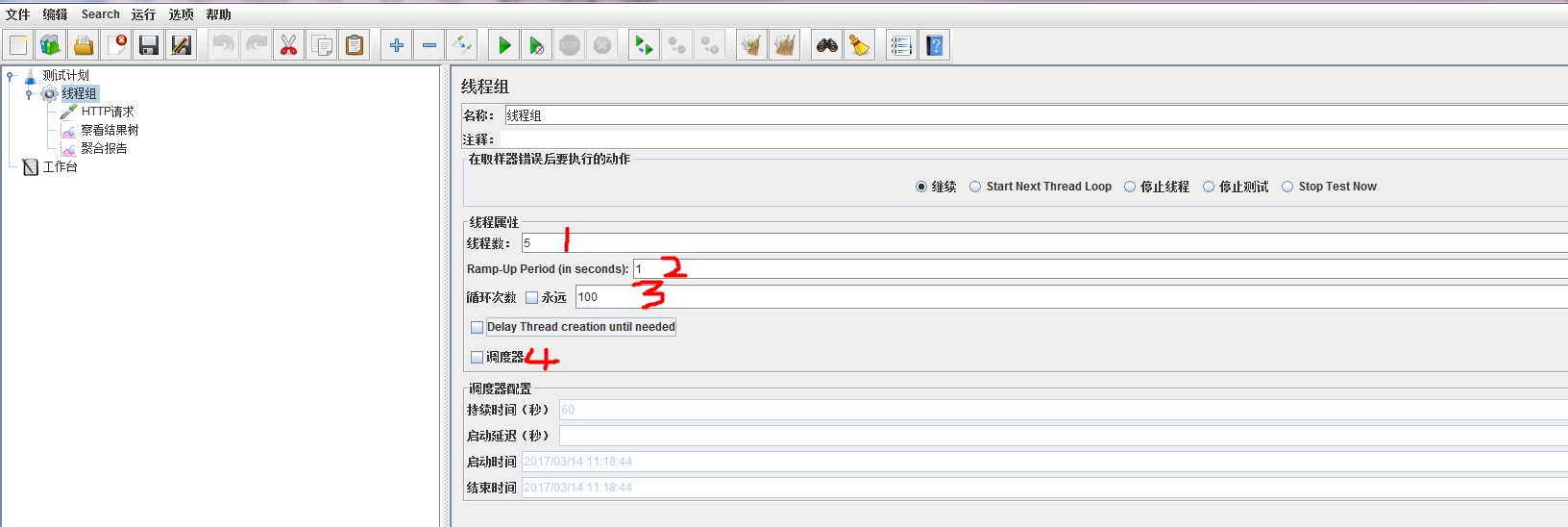
在“响应数据”下，我们看到了与使用浏览器一样的返回结果（报文中headers有差异，这部分可以忽略，想知道为什么不一样的可以去学学http协议，在这里不影响结果）。

到这里，一个http接口测试的Jmeter脚本已经完成。“Ctrl+S”保存一下吧，下次就可以直接拿来使用了（Jmeter脚本默认保存在bin目录下）

## 压力测试

在上面脚本配置上线程数，即可对该接口进行压力测试啦。

* 点击“线程组”



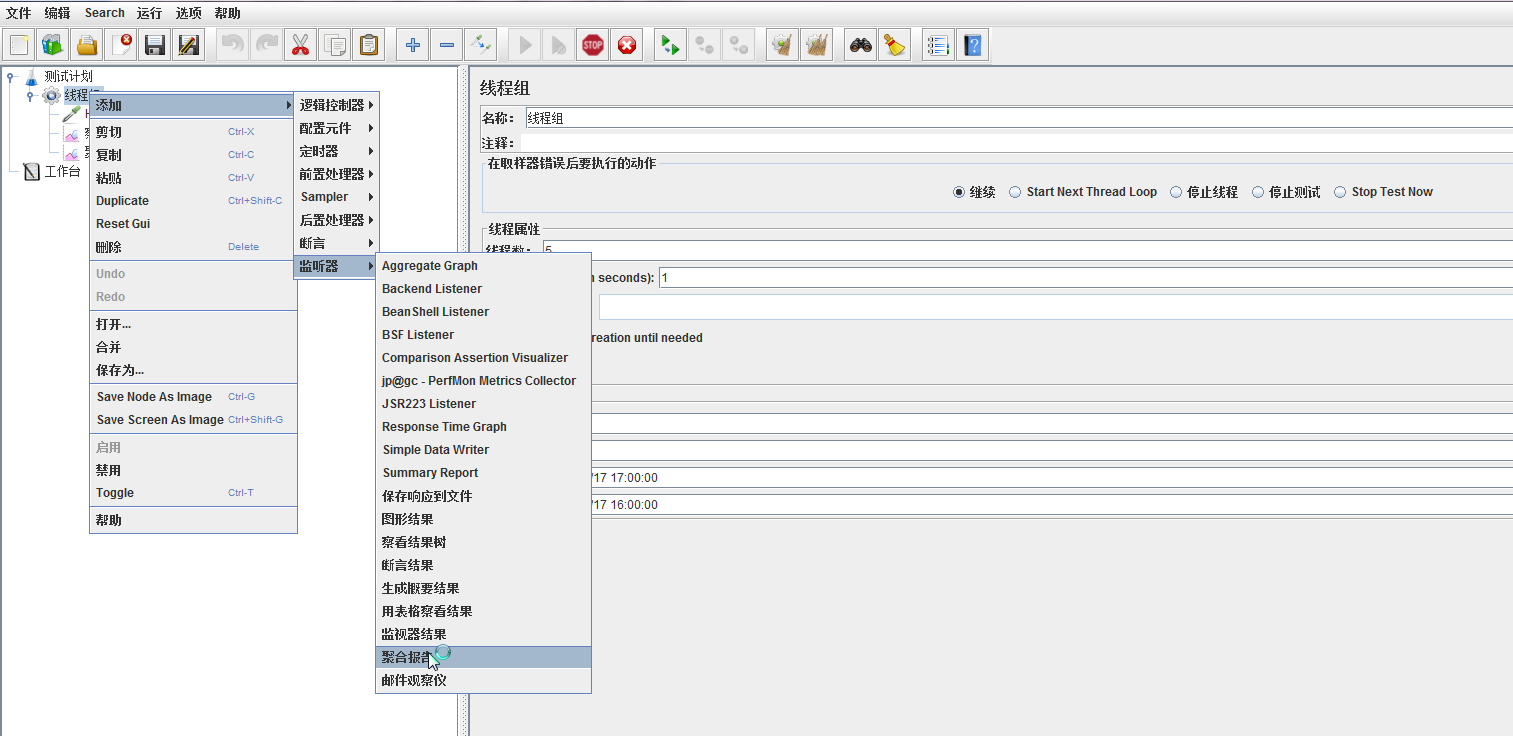
1. 线程数：需要采用多少个并发。
2. Ramp-Up Period(in seconds):设置到达最大并发所需时间（如上图，从0增加到5个并发，花费1秒钟）
3. 循环次数：每个并发执行几遍脚本（如上图，每个线程执行100遍）。勾选永远，则脚本会被无限循环执行下去。
4. 调度器：勾选则启用调度器

持续时间（秒）：设置脚本跑多久（如需要让脚本跑一分钟，则填入60）

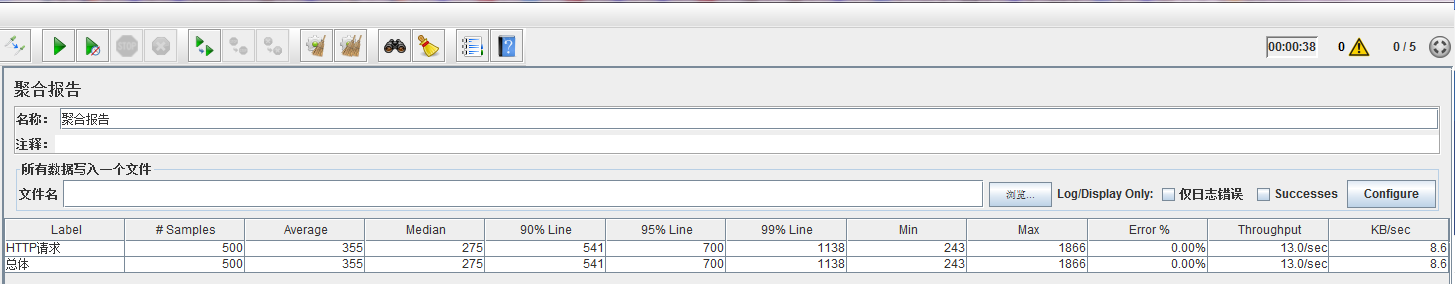
启动延迟（秒）：延迟多少秒才开始执行脚本

* 添加“聚合报告”

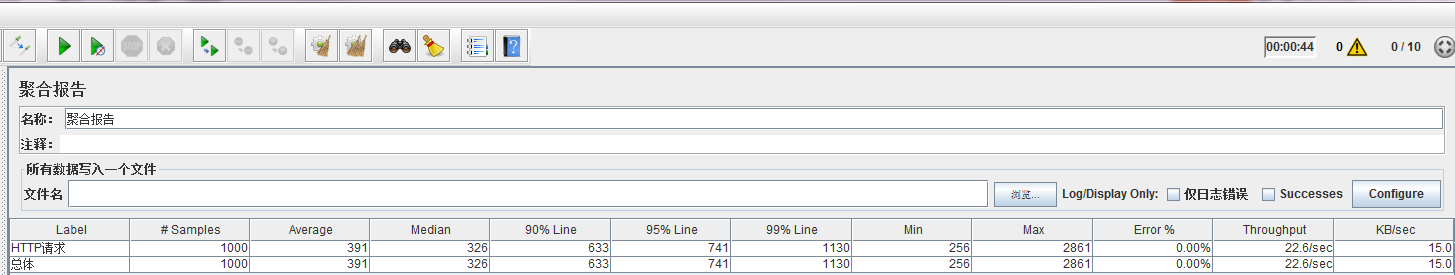
聚合报告是一个监听器，作用是，为本次压力测试整理出一份“报表”



* 点击按钮让我们来看看5并发循环100次的结果



* 点击按钮，清除上次测试结果，然后来看看10并发循环100次结果



可以看到5并发时，Throughput为13.0/sec；10并发时：Throughput为22.6/sec。

理论上，并发数与吞吐量成正比，并发越多，吞吐量越大。

实际上，服务器的处理能力和网络带宽是有限的，当并发数达到一定数量之后，吞吐量不随并发数增加而增长，出现响应时间变长，甚至服务器宕机等状况，称之为服务器的性能“瓶颈”。

简单来说，性能测试就是根据服务器的“瓶颈”，来评估服务的性能是否能够达到期望。

附：表头说明

*Label - 如图，在不勾选"Include group name in label?"复选框的情况下，为请求取样器的名称，否则为“请求取样器所在线程组:请求取样器名称”*

*# Samples - 用同一个请求取样器，发送请求的数量(注意：该值是不断累计的)。比如，10个线程数设置为10，迭代10次，那么每运行一次测试，该值就增加10\*10=100*

*Average - 默认情况下是单个Request的平均响应时间，当使用了Transaction Controller 时，也可以以Transaction为单位显示平均响应时间*

*Median - 中位数。表示响应时间本不大于该时间值的请求样本数占总数的50%*

*90% Line - 表示响应时间不大于该时间值的请求样本数占总数的90%*

*Min - 针对同一请求取样器，请求样本的最小响应时间*

*Max - 针对同一请求取样器，请求样本的最大响应时间*

*Error % - 出现错误的请求样本的百分比*

*Throughput - 吞吐量以“requests/second、requests /minute、requests /hour”来衡量。 时间单位已经被选取为second，所以，显示速率至少是1.0，即每秒1个请求。 当吞吐量被保存到CVS文件时，采用的是requests/second，所以30.0 requests/second 在CVS中被保存为0.5Kb/sec - 以Kilobytes/seond来衡量的吞吐量*

## 附上演示脚本

上面演示步骤最后保存的脚本，可以直接用Jmeter打开，运行。



# 录制HTTP脚本

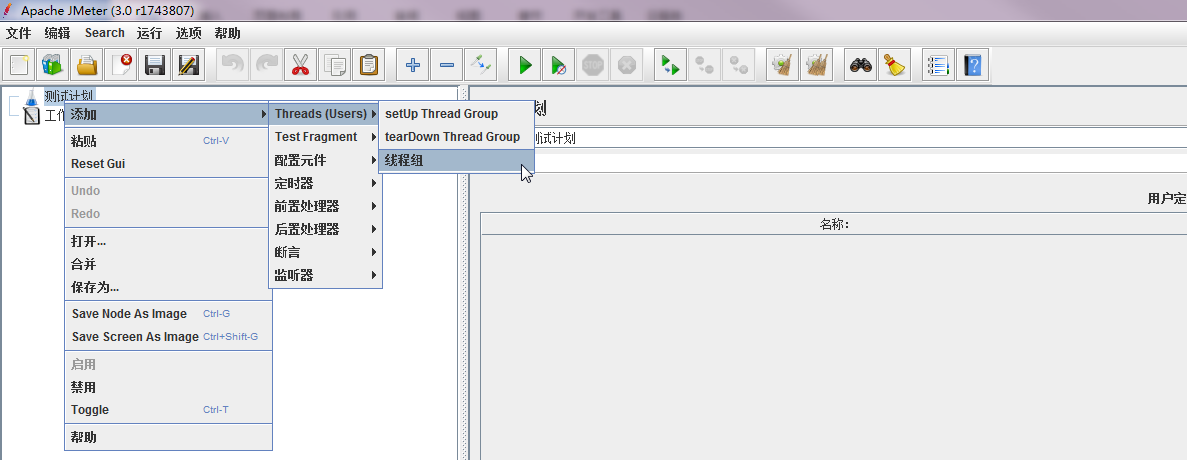
当需要为多个接口做脚本时，手动编制就显得太麻烦，这时候我们可以使用Jmeter自带的脚本录制功能。

录制原理是Jmeter启用一个代理服务器，然后配置浏览器代理，Jmeter就可以对浏览器进行抓包，然后生成脚本。

## Jmeter配置代理服务器并录制脚本

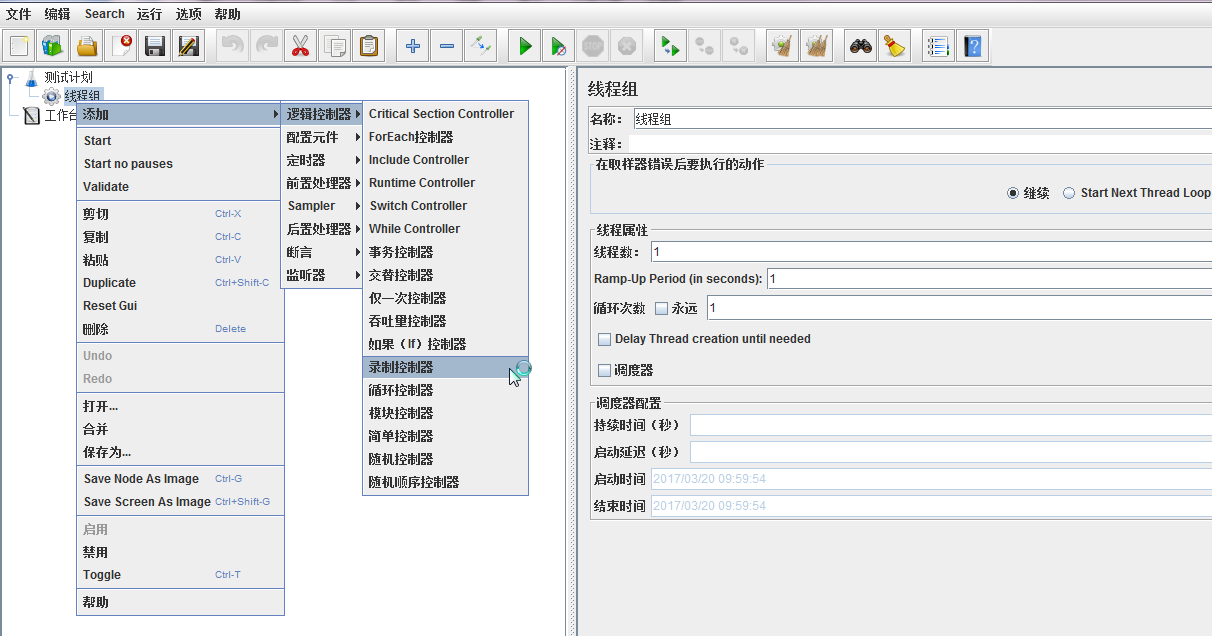
### 添加线程组

* 右键“测试计划”->“添加”->“Threads(Users)”->“线程组”



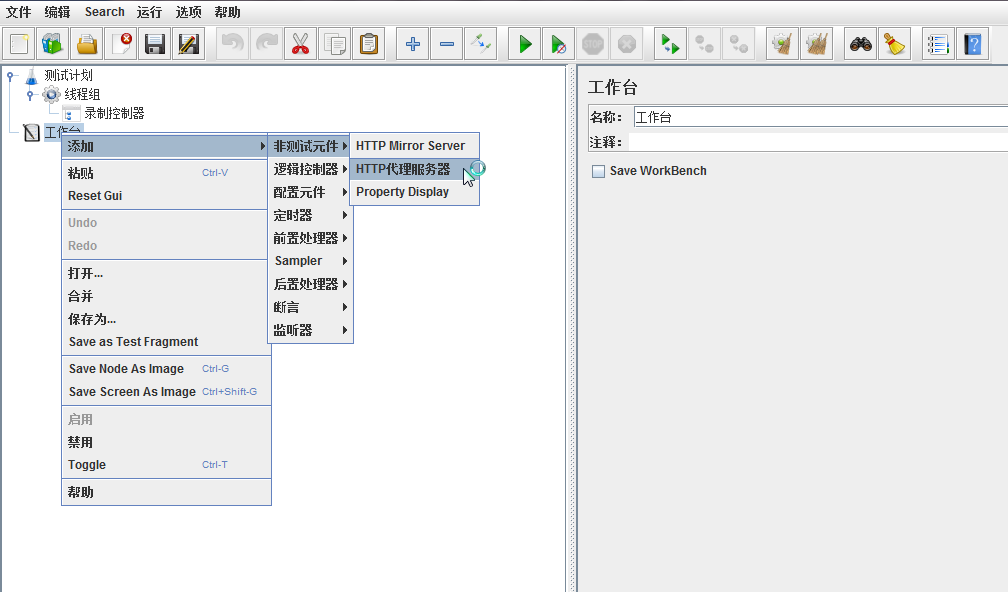
### 添加录制控制器

* 右键“线程组”->“添加”->“逻辑控制器”->“录制控制器”

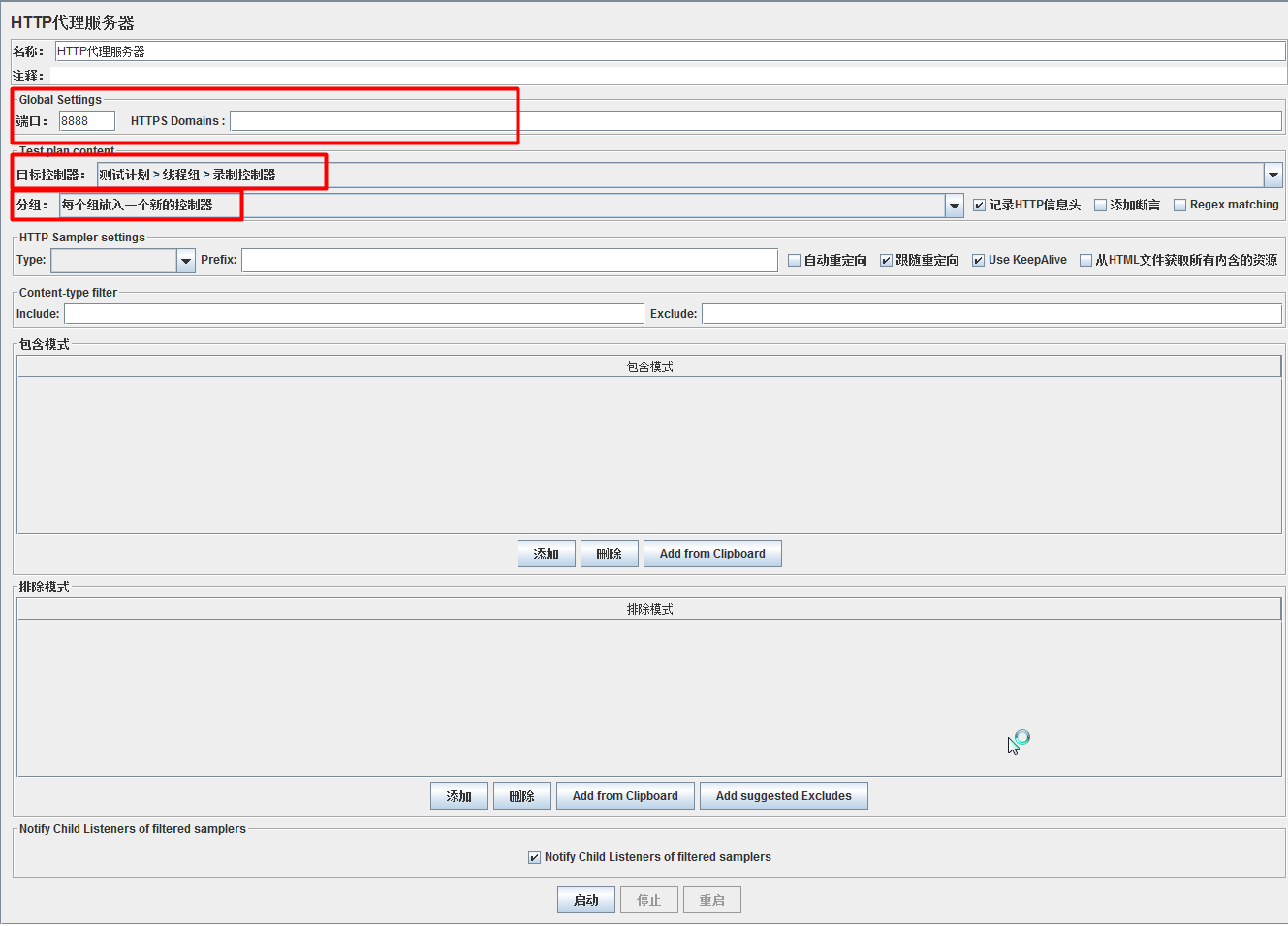


### 添加代理服务器

* 右键“工作台”->“添加”->“非测试元件”->“Http代理服务器”



* 配置代理服务器

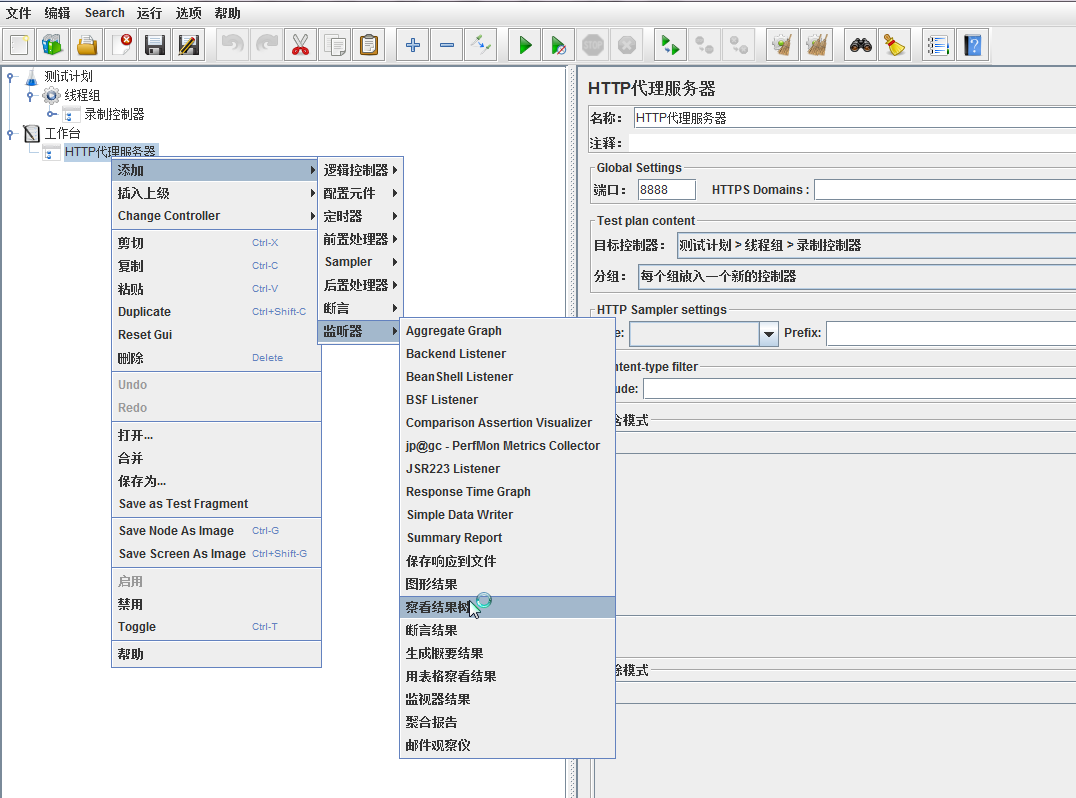


1. 端口：用默认的8888就好，如果冲突，可以换一个不冲突的端口号。
2. HTTPS Domains：默认本机localhost
3. 目标控制器：选择 测试计划>线程组>录制控制器
4. 分组：选择 每个组放入一个新的控制器

然后点击“启动”，弹出的提示框点击“确定”，代理就启动成功了。

* 添加“察看结果树”

右键“HTTP代理服务器”->“添加”->“监听器”->“察看结果树”



### 设置浏览器代理

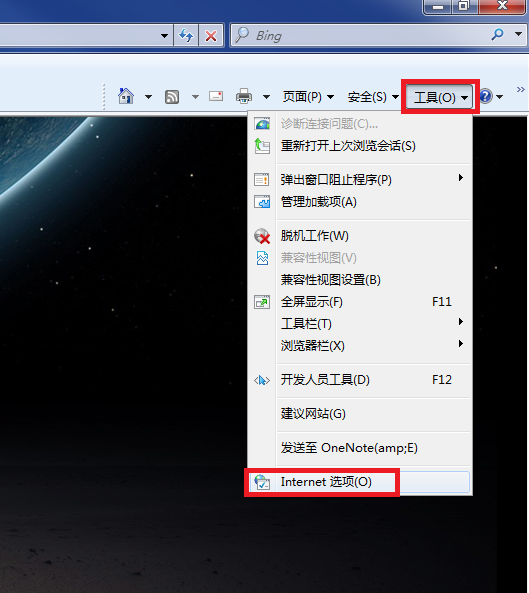
Jmeter代理服务器启动之后，再设置浏览器代理，就可以抓包了。

不同浏览器的代理设置方式不一样。一般来说，浏览器都是在设置系统代理，一个浏览器设置了代理，其它浏览器也会用这个代理。

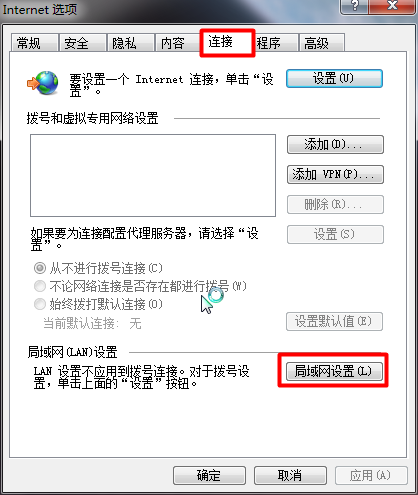
下面介绍三种常用浏览器代理的设置方法。

#### IE浏览器

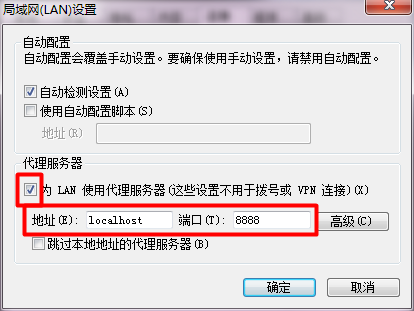
1. 点击“工具”->“Internet选项”



1. 在新打开的窗口中选择“连接”，点击“局域网设置（L）”



1. 在弹出窗口，选中“为LAN使用代理服务器（这些设置不用于拨号或VPN练级）（X）”，然后在“地址（E）”填入localhost，“端口（T）”填入刚设置的代理端口号8888。然后点击“确定”完成配置。

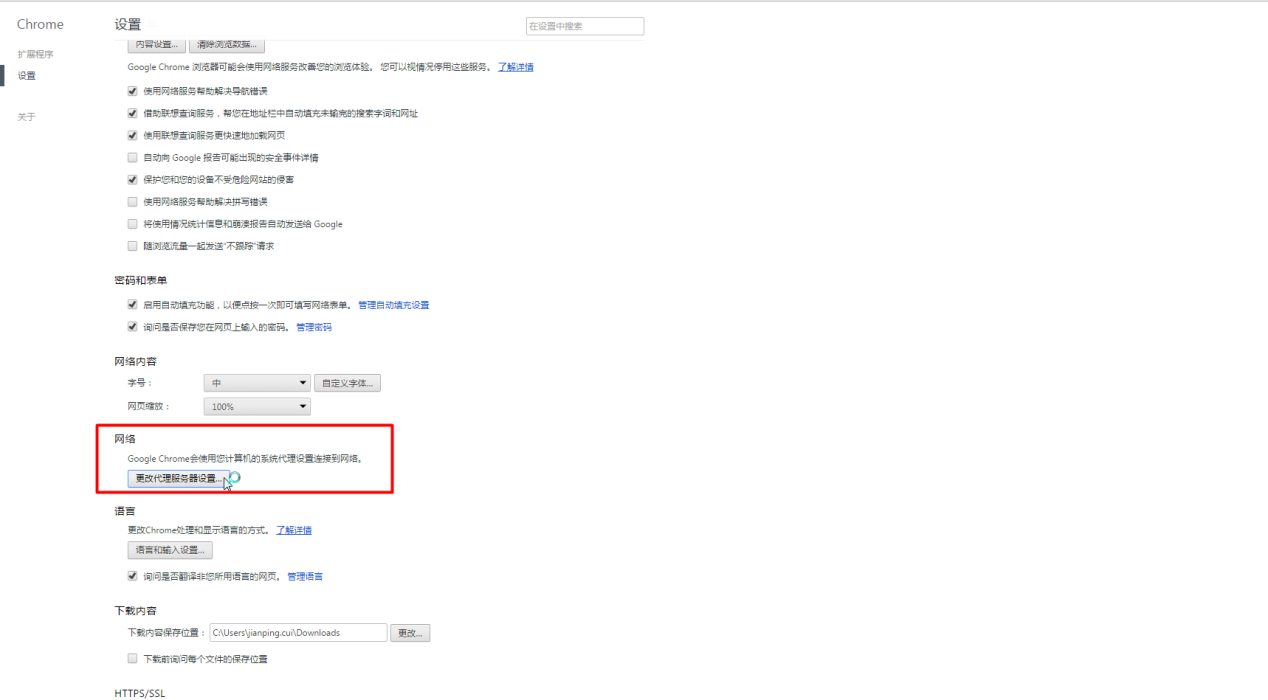


#### Google Chrome（谷歌浏览器）

1. 点击右上角，展开菜单，点击“设置”



1. 点击最下方“显示高级设置...”，找到“更改代理服务器设置...”，点击



1. 弹出新窗口后，剩下的操作照着IE浏览器设置代理的2、3就可以了

#### Firefox（火狐浏览器）

1. 点击右上角，展开菜单，点击“选项”

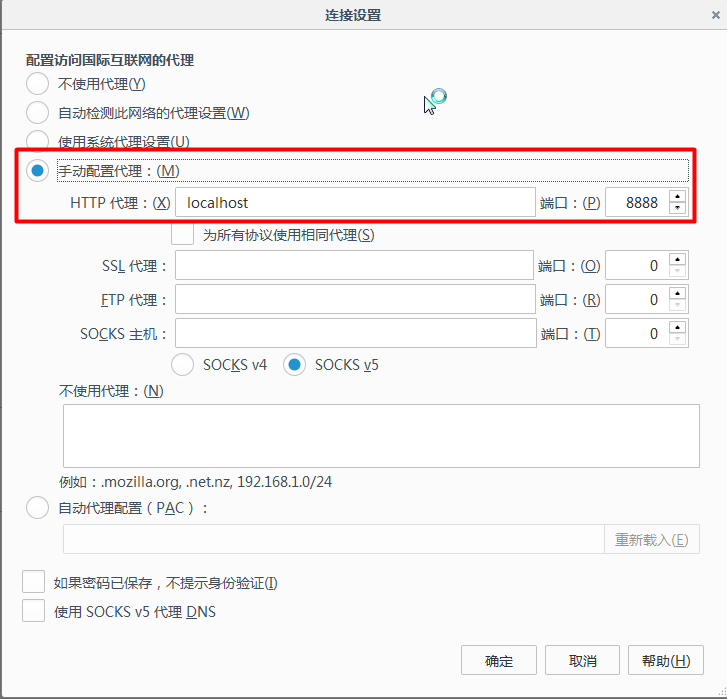


1. 选择“高级”->“网络”，点击“设置（E）...”



1. 在新窗口配置代理：选择“手动配置代理:（M）”，“HTTP代理:（M）”填入“localhost”，“端口:（P）”填入“8888”，点击“确定”，设置完成。

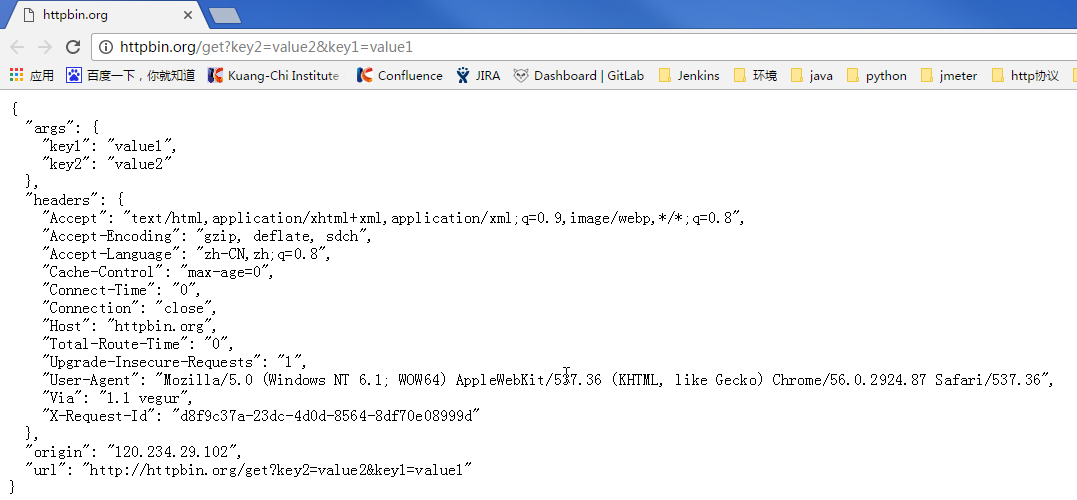
Tip：这里设置的代理是设置的自己的，不会影响系统代理设置。



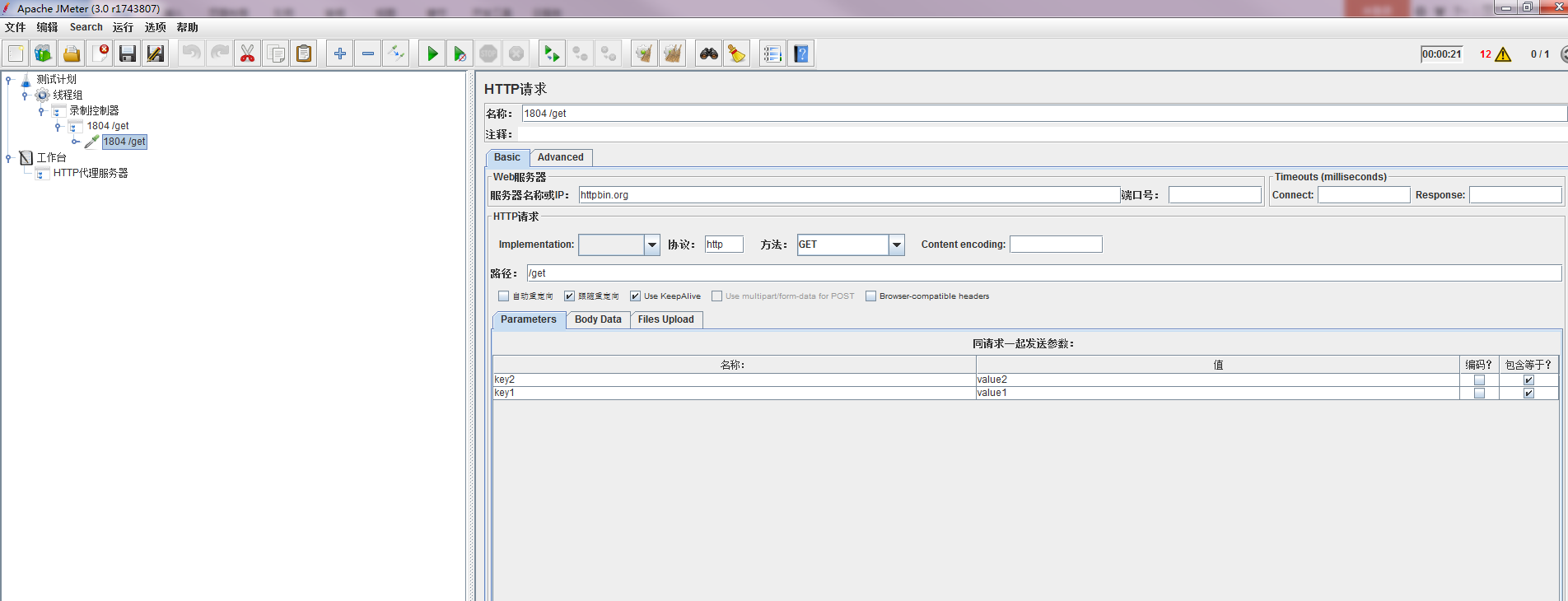
### 录制脚本

所有准备工作都做完，可以开始录制脚本了，还用上面举例的接口，这里尝试录制该接口的脚本。

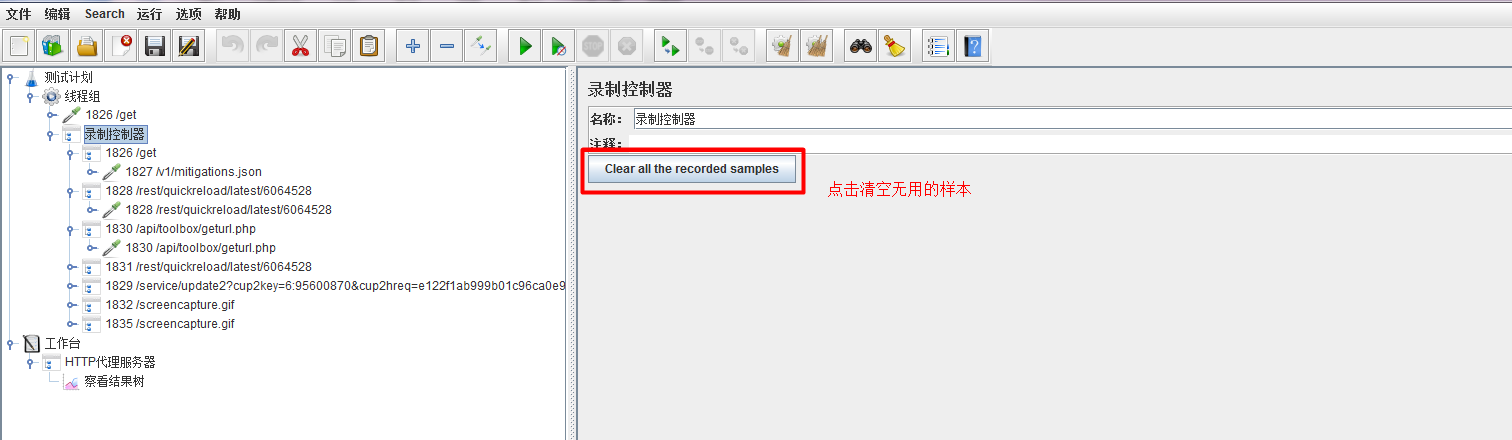
1. 在一个设置好代理的浏览器中打开该接口，用Google Chrome演示一下



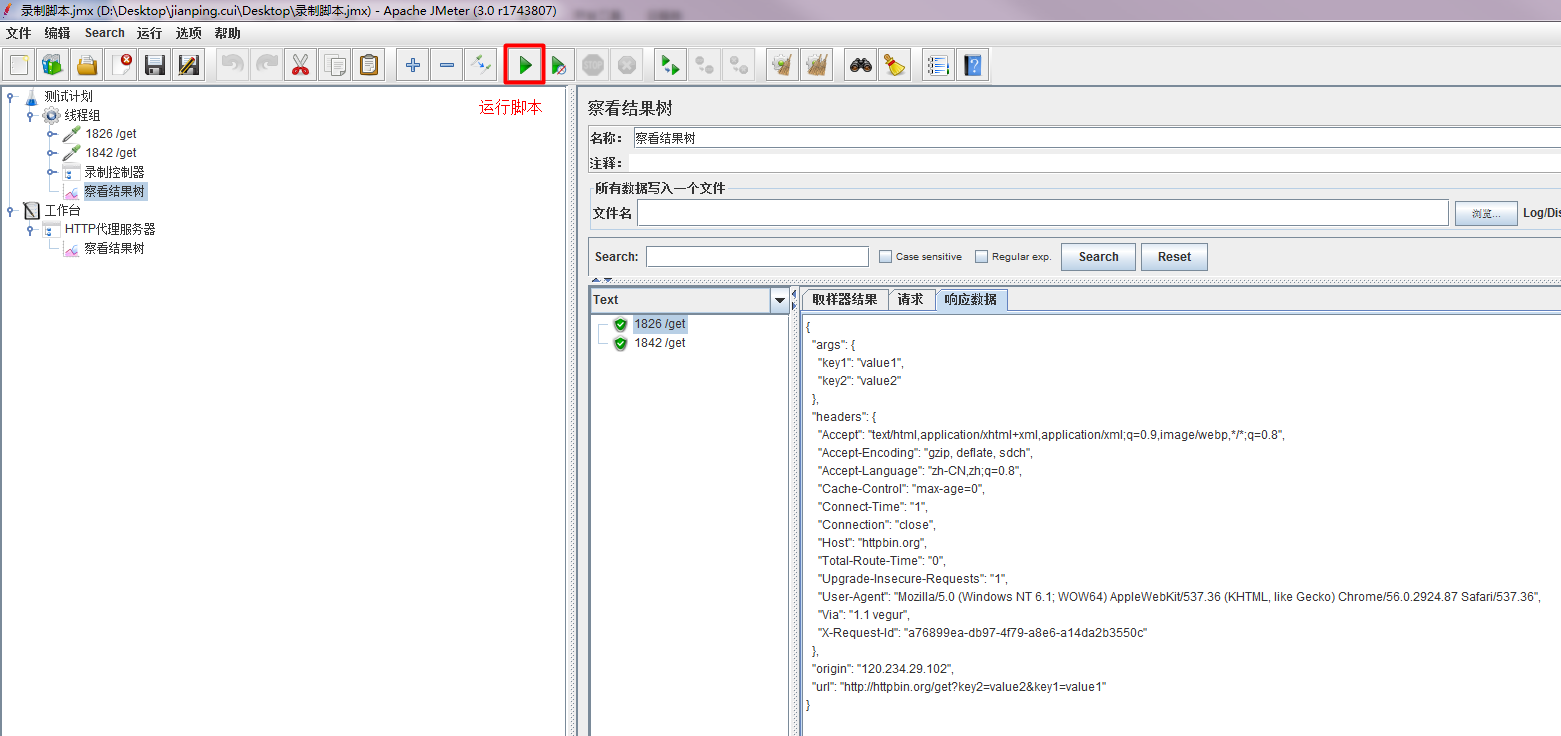
1. 然后我们看看Jmeter，这里捕获了这次请求的信息，并在“录制控制器”下生成了一条Http请求，如下



1. 将需要的Http请求拉倒“线程组”下，然后清空“录制控制器”，然后就可以开始下一个接口的录制



1. 录制完成之后，在线程组下添加“察看结果树”，“Ctr+S”保存脚本，运行一下，看到两条请求都成功了，一个脚本就录制成功了！



### 一个未解决的问题

在录制脚本时，会发现很多无用的请求干扰，在这里Jmeter代理提供了一个过滤器，如接口是\*.html的，使用“.\*/\*.html”过滤。

这里配置了过滤，总不好用，要么什么也没过滤掉，要么都过滤没了，还有把需要的一部分也过滤没了。

鉴于此，录制脚本时，尽量少开其它网页，减少干扰。

如有哪位知道怎么设置这个过滤器，请告知，感激不尽。

# TCP测试

就像简介上说的，Jmeter不单可以做Web测试，也可以TCP测试。TCP脚本跟HTTP脚本做成十分类似，但是有些地方会不同，还是啰嗦介绍一下。

## 创建一个TCP测试桩

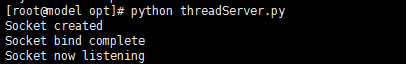
先创建一个简单的TCP服务器，用来学习Jmeter的TCP测试，这个服务器只做一个事情，就是收到一个报文，然后给一个响应。



这是从网上找的一份简单的TCP服务器代码，python语言写的，直接运行就好，端口是8888，IP则为运行该脚本的机器的IP地址。

* 问题：怎么运行这个服务器？

答：有很多方法可以启动，介绍一种简单的，找一个Linux服务器，上传这个文件，然后执行命令python threadServer.py就OK了。

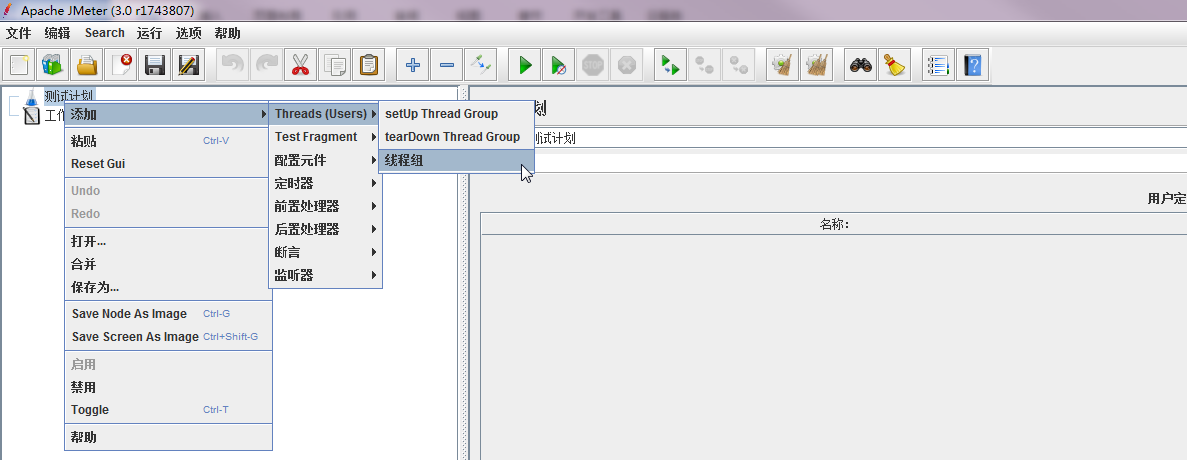


## 创建TCP测试脚本

和HTTP测试脚本类似

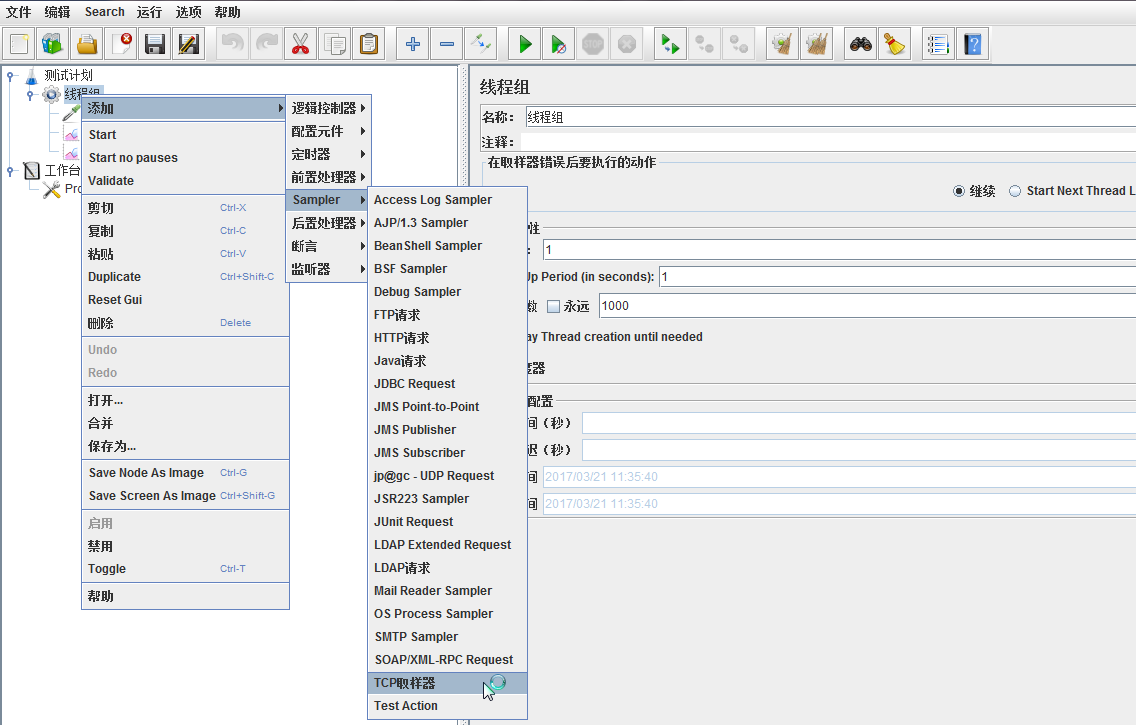
### 添加线程组

* 右键“测试计划”->“添加”->“Threads(Users)”->“线程组”

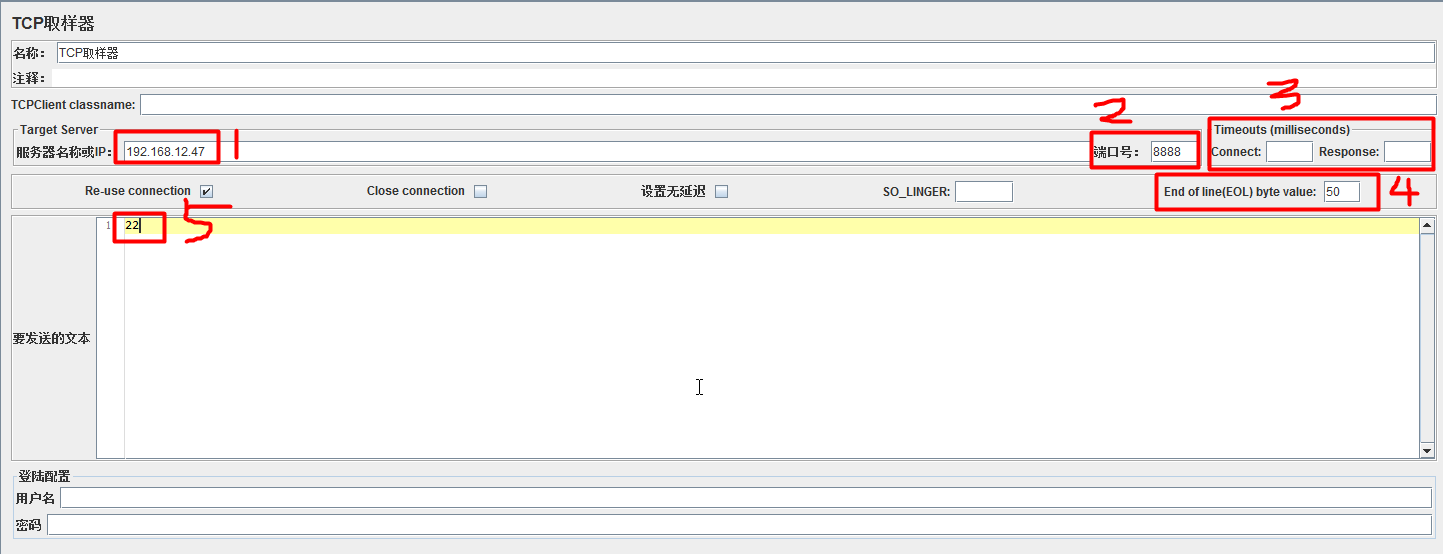


### 添加“TCP取样器”

* 右键“线程组”->“添加”->“Sampler”->“TCP取样器”



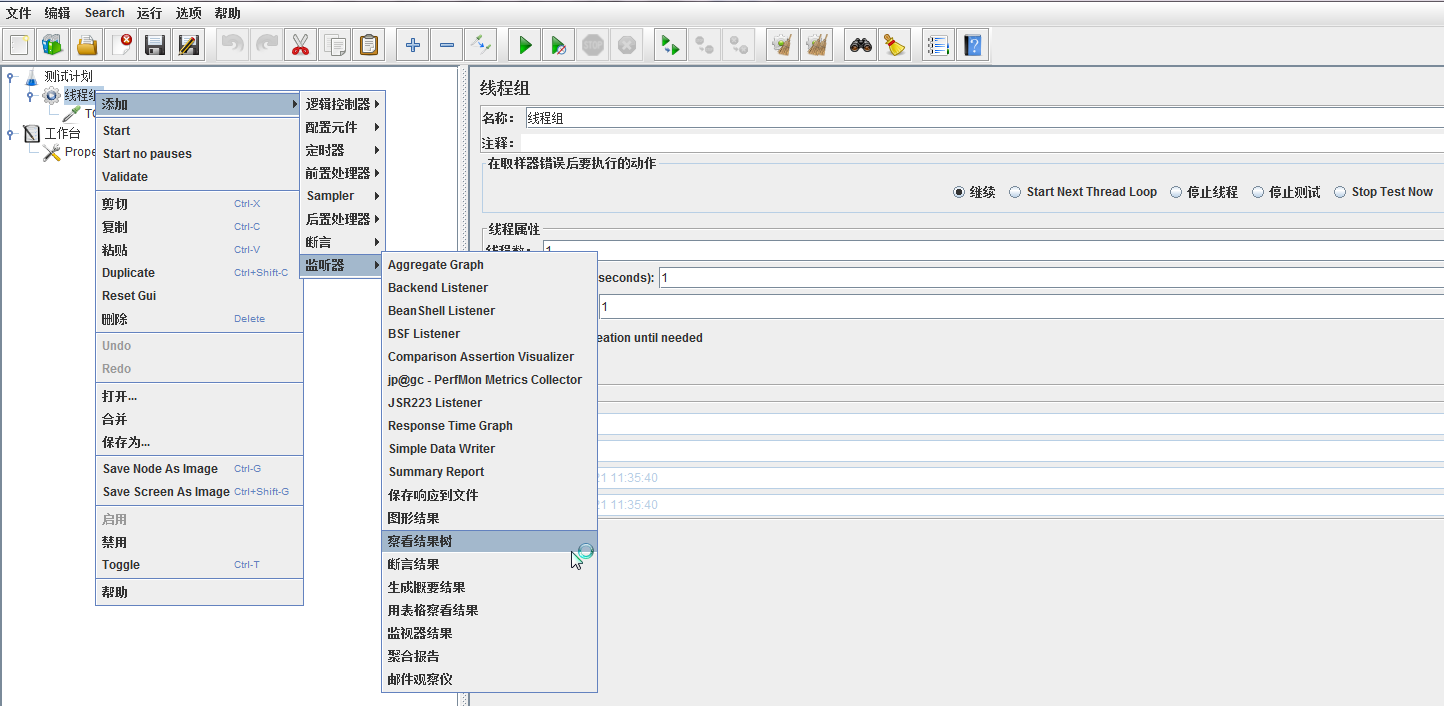
* 看下“TCP取样器”的参数



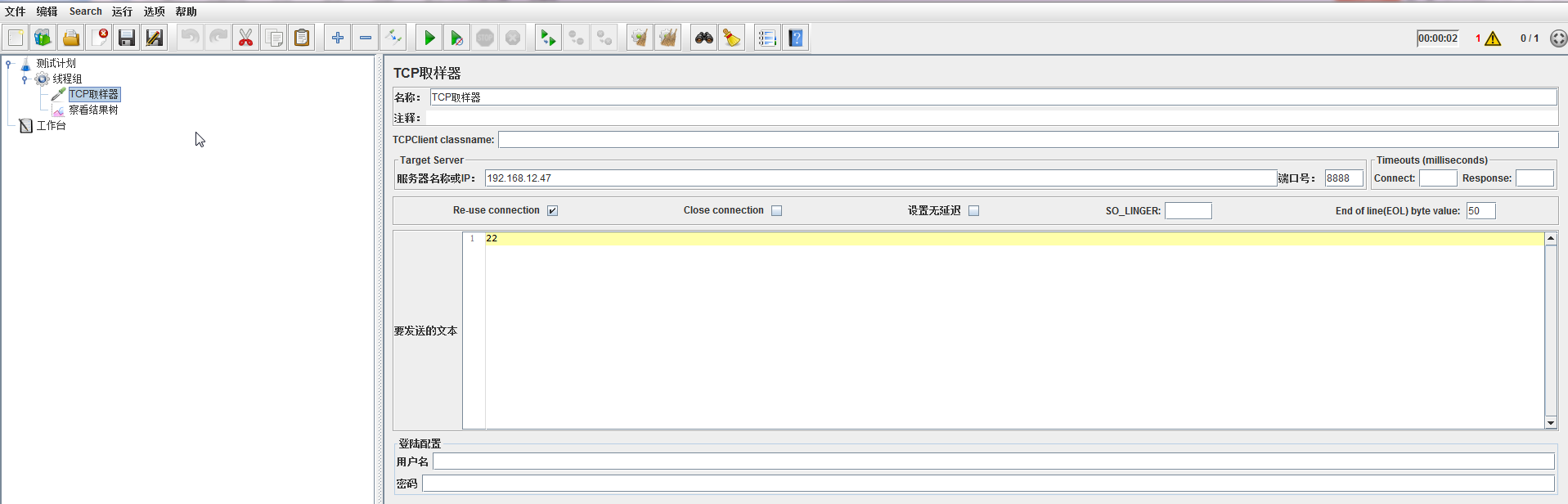
1. 这里填服务器IP
2. 端口号
3. 超时时间，左边连接超时，右边响应超时，单位都是毫秒
4. EOL，每个最后一个字符的byte值，这个是用来校验接收报文是否结束
5. 这里是需要发送的报文

### 添加监听器

* 右键“线程组”->“添加”->“监听器”，选择“察看结果树”

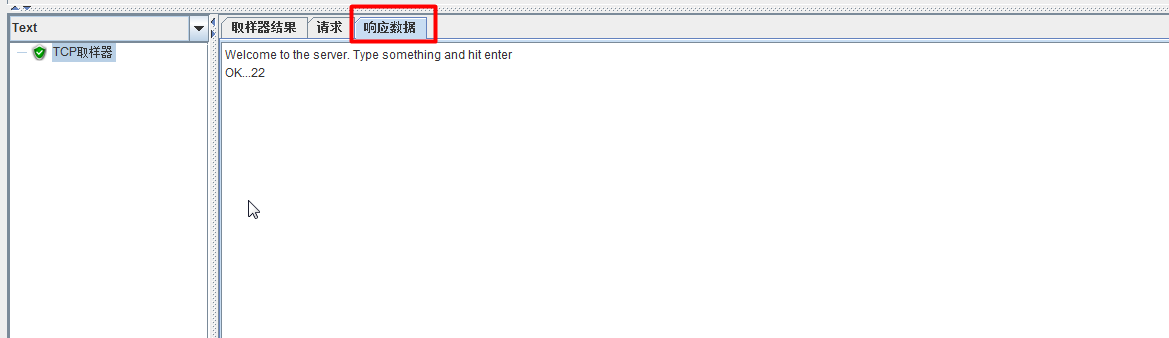


到这一步，一个TCP测试脚本就完成了，整体效果如下



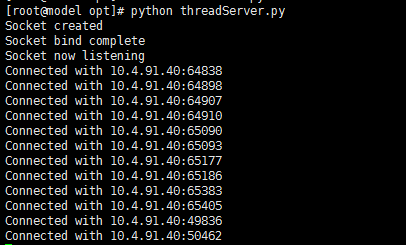
### 运行脚本

* 点击按钮，运行脚本，察看结果

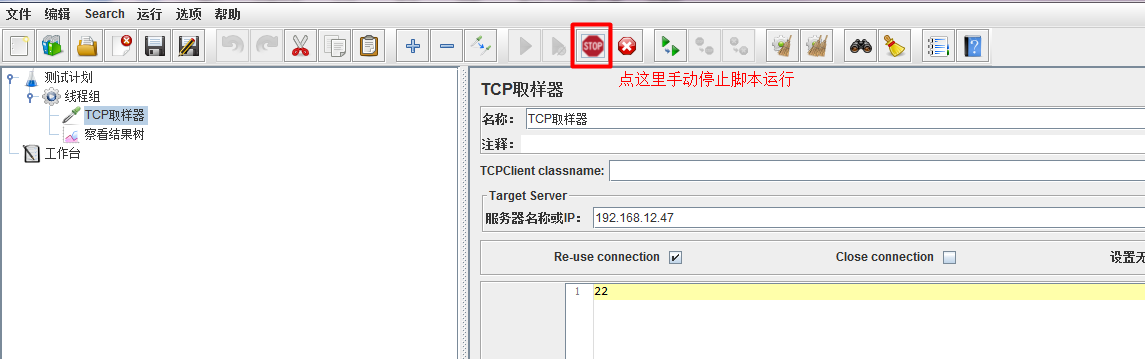


右边框里的就是TCP服务器响应报文，脚本这就跑通了

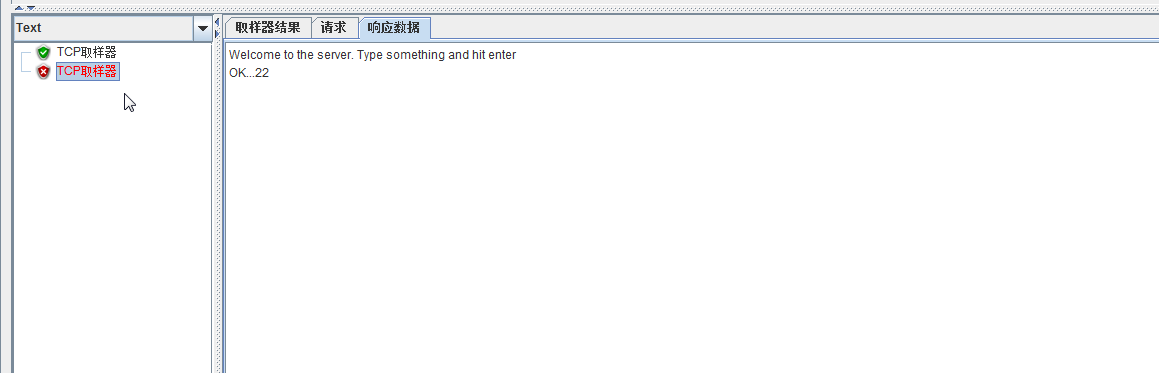
服务器后台响应



注：这里对EOL特殊说明一下，第一次运行脚本，不知道这个EOL填写什么，可以先运行脚本，然后手动停止运行脚本，如下



然后见到如下结果



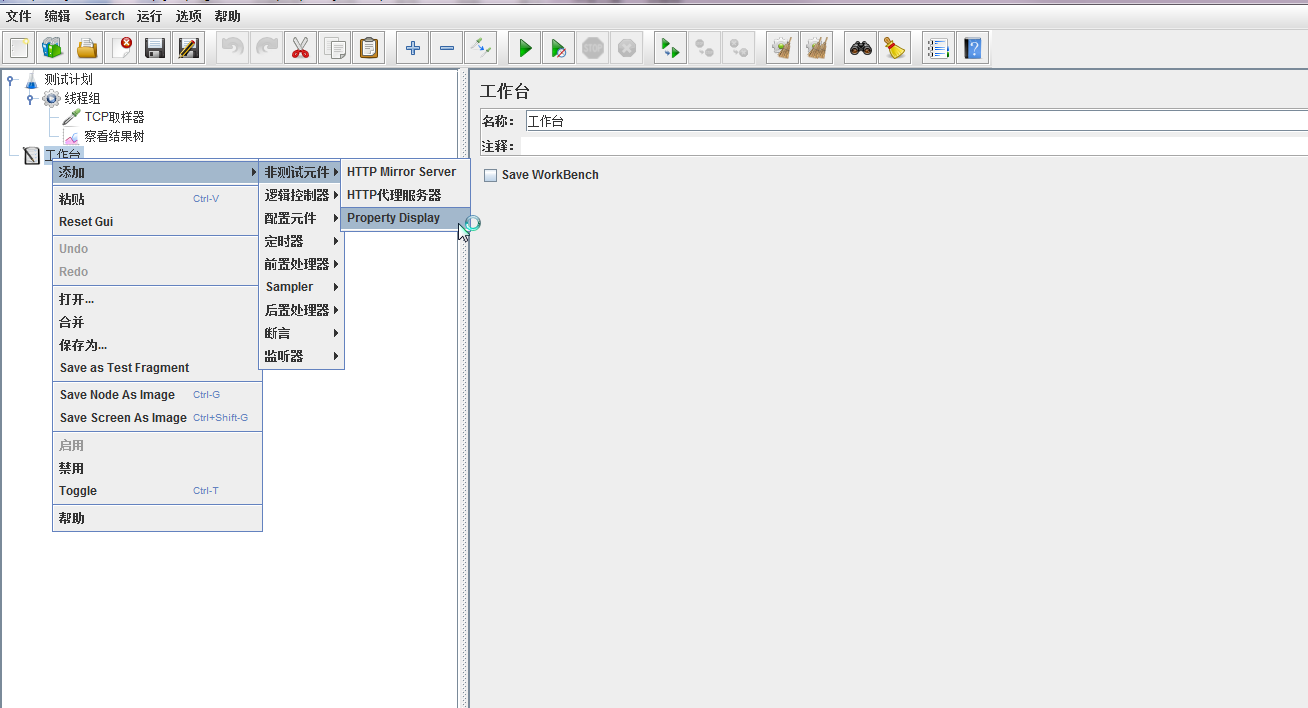
脚本执行失败，但是我们仍然获得了响应报文，将最后一个字符‘2’转换为byte值就是50，所以之前的EOL填入50。

### 发送16进制字节流报文

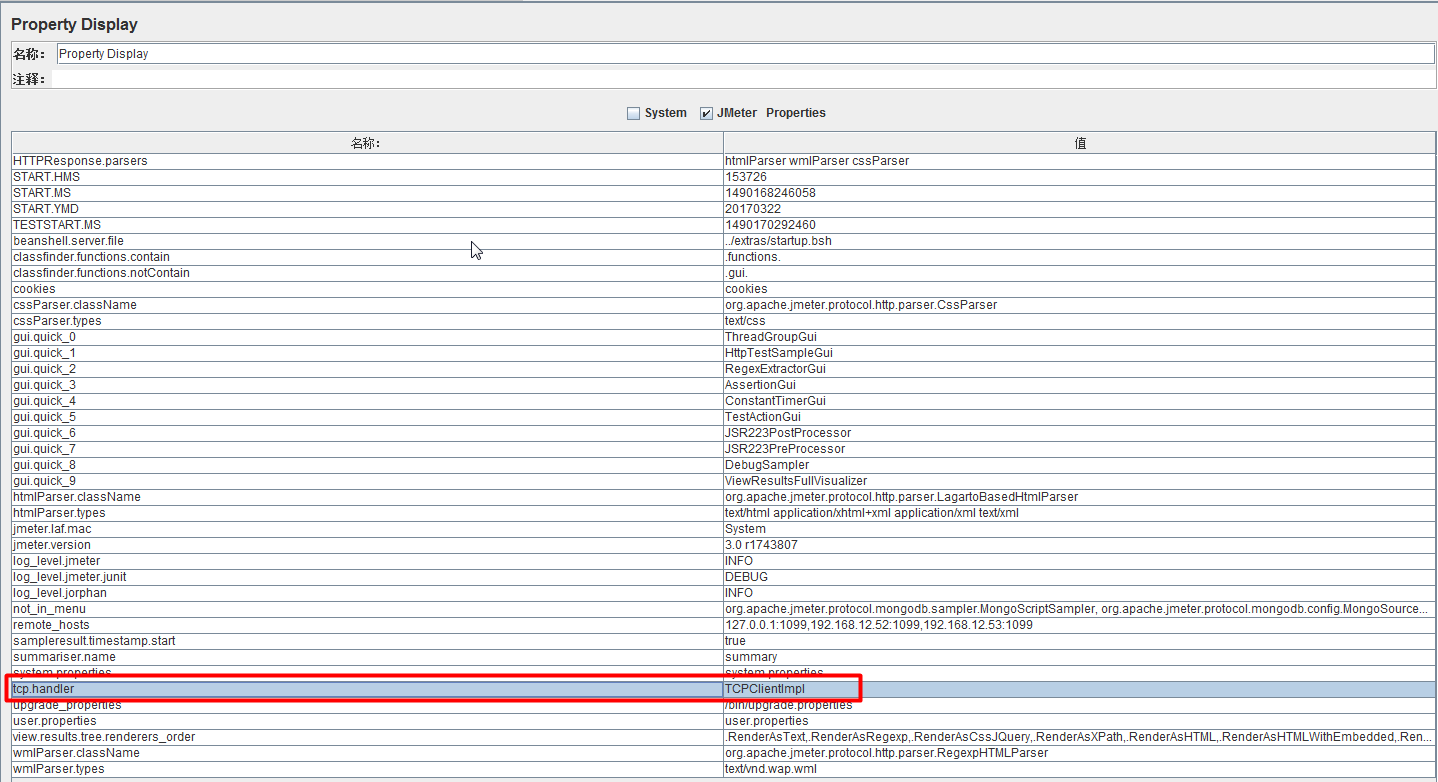
上面介绍的都是工具默认的文本报文，Jmeter同样支持发送16进制报文，我们只需要修改一个参数值就好了。

* 修改Jmeter参数

右键“工作台”->“非测试元件”->“Property Display”



找到参数tcp.handler，如下



修改其值为BinaryTCPClientImpl就OK了，如果发现这个参数没有生效，按F5刷新一下。

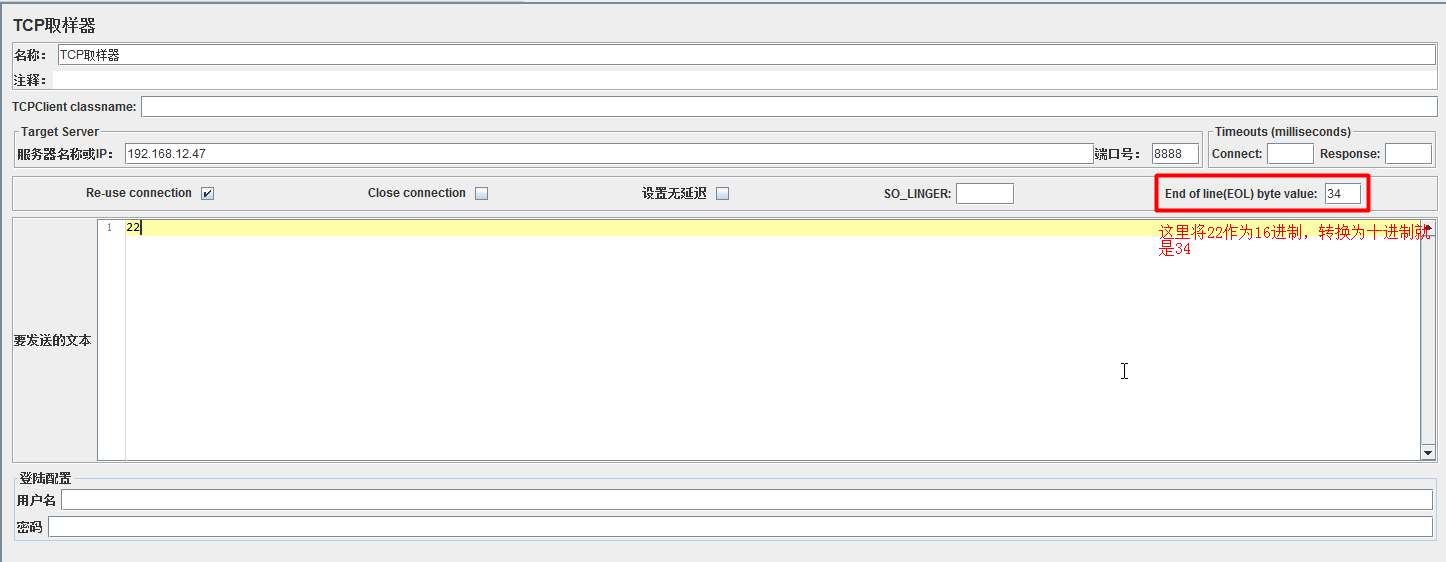
Tips：

1. tcp.handler有三种设置：

TCPClientImpl：文本数据  
 BinaryTCPClientImpl：传输二进制数据，指定包结束符。  
 LengthPrefixedBinaryTCPClientImpl：数据包中前2个字节为数据长度。

1. 上面介绍的修改参数是临时的，重启Jmeter之后就会还原，需要永久修改的可去修改bin目录下的配置文件jmeter.properties中的参数tcp.handler，重启Jmeter生效。

* 发送16进制字节流报文，脚本需要做如下调整



# UDP测试

Jmeter做UDP测试，需要下载插件支持，这里先不介绍Jmeter插件安装，上面提供的Jmeter已配置好插件。

## 创建一个UDP测试桩

这里用python做了一个简单的udp服务器，该服务器只会一件事情，就是将收到的报文内容返还给你。



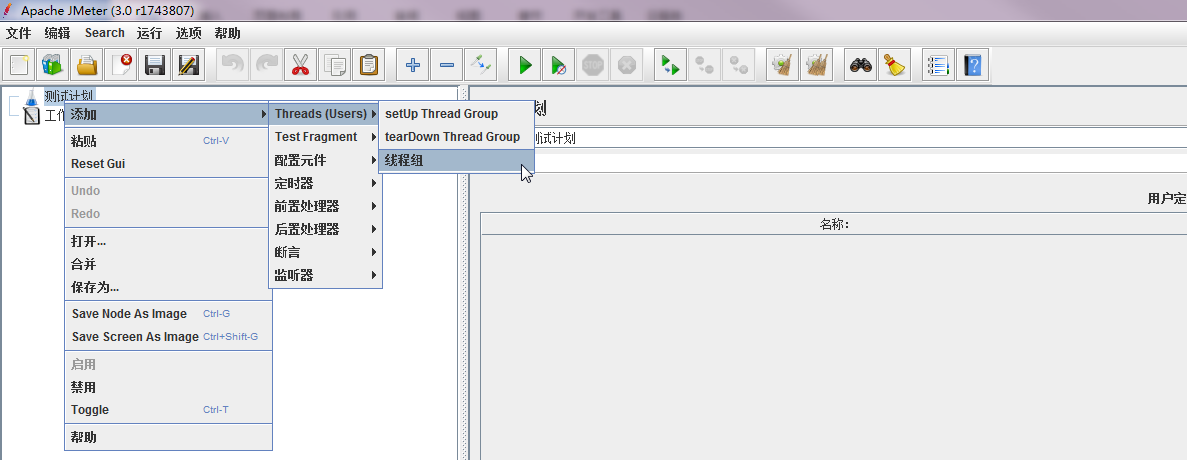
运行该服务器的方法同5.1一样，找一个Linux服务器，上传该python文件，执行命令“python udpServer.py”，该服务器端口为54321，IP为Linux服务器的IP地址。



## 创建UDP测试脚本

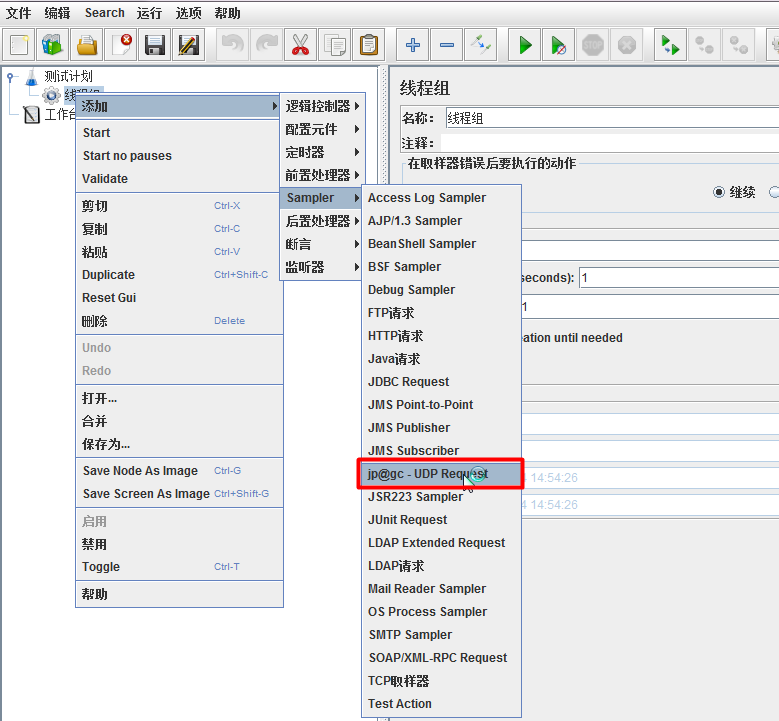
### 添加线程组

* 右键“测试计划”->“添加”->“Threads(Users)”->“线程组”

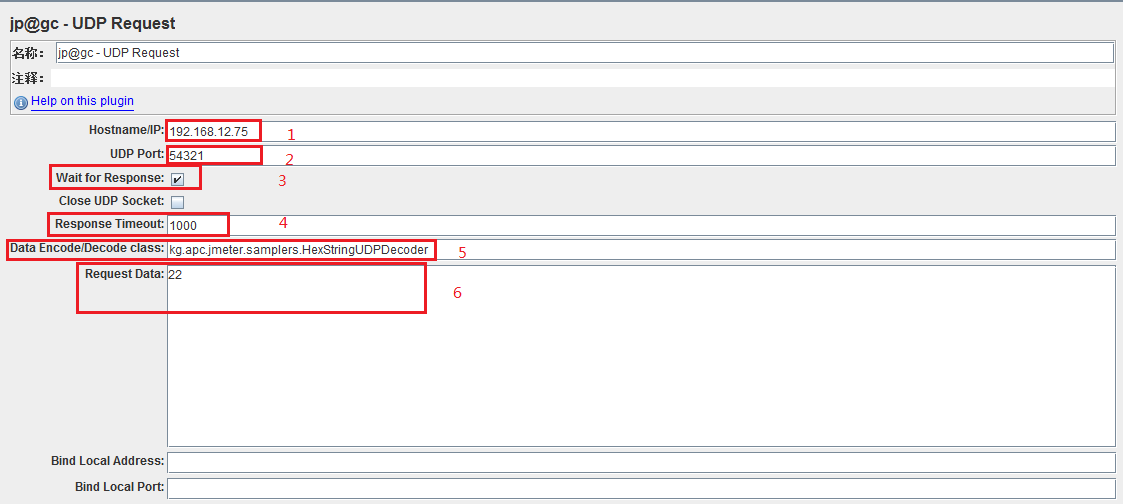


### 添加“jp@gc - UDP Request”

* 右键“线程组”->“添加”->“Sampler”->“jp@gc - UDP Request”



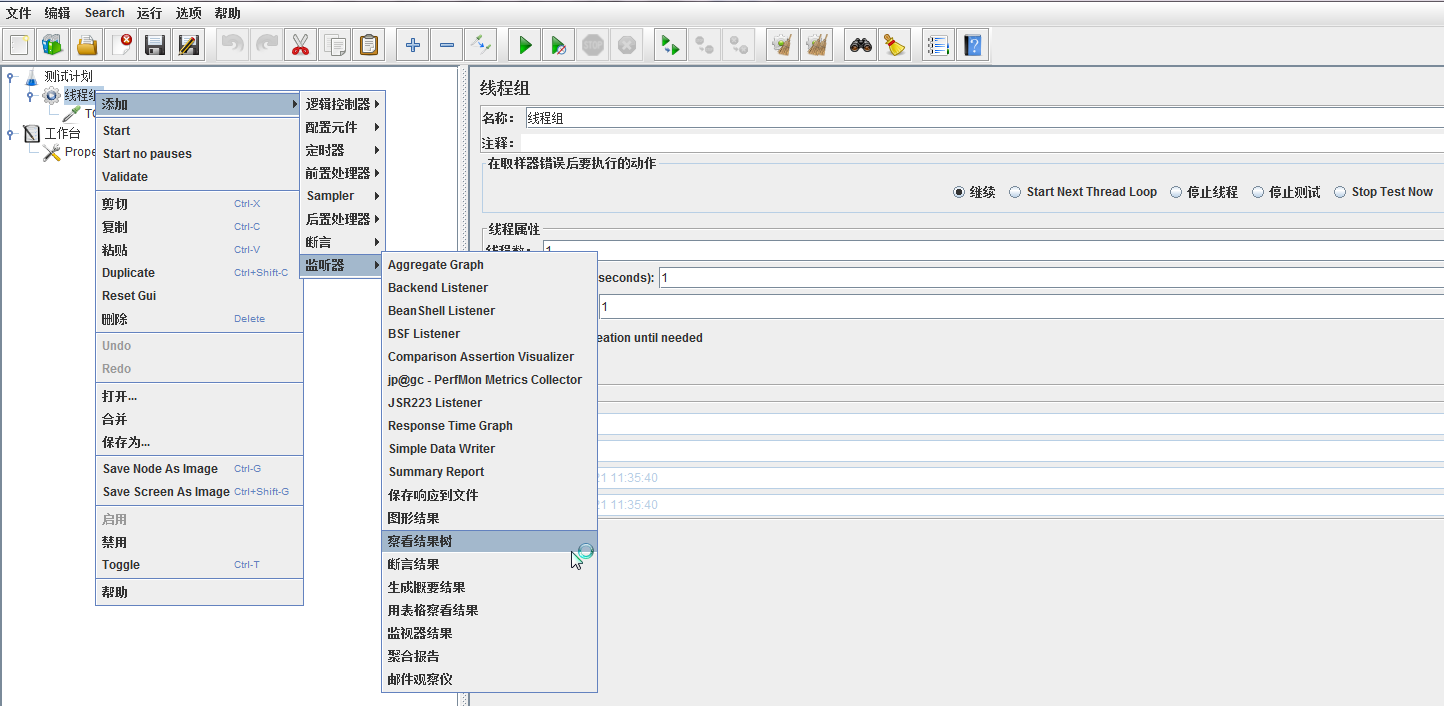
* 配置“jp@gc - UDP Request”



1. 服务器IP地址
2. UDP服务器端口号
3. 勾选等待返回报文，不勾选则只管发送报文，不接收
4. 等待返回报文的超时时间
5. 发送报文的编码格式，这里可以手动编辑  
    kg.apc.jmeter.samplers.HexStringUDPDecoder 直接发送16进制数据，HEX-encoded  
    kg.apc.jmeter.samplers.UDPSampler 填写字符串  
    kg.apc.jmeter.samplers.DNSJavaDecoder dns解析填写格式如: www.6san.com. A IN  
    kg.apc.jmeter.samplers.UDPTrafficDecoder接口可以自定义编码/解码
6. 需要发送的报文

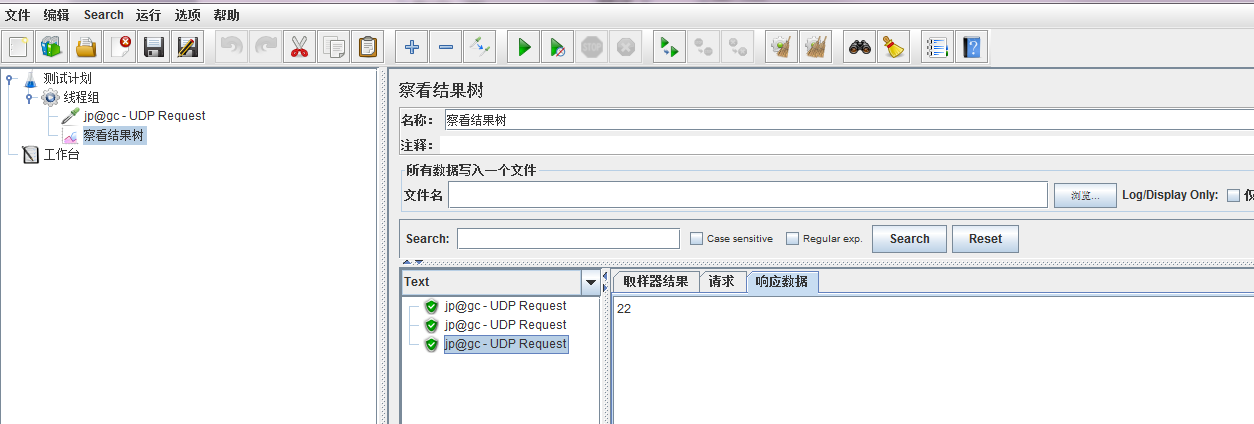
### 添加监听器

* 右键“线程组”->“添加”->“监听器”，选择“察看结果树”



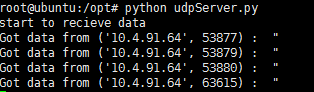
### 运行脚本

* 点击按钮，运行脚本，察看结果



这里看到，Jmeter已经接收到了UDP服务器的返回值

服务器后台也有简单日志打出来



### 设置UDP报文格式

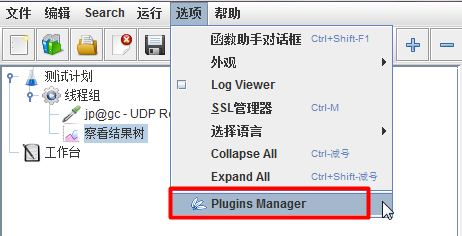
修改字段“kg.apc.jmeter.samplers.UDPSampler”的值，默认发送16进制字符流。

如果想发送字符串，则将该字段值改成kg.apc.jmeter.samplers.UDPSampler

# Jmeter分布式部署

# 配置Jmeter插件管理

1. 上官网https://jmeter-plugins.org/downloads/all/
2. 下载插件管理软件[plugins-manager.jar](https://jmeter-plugins.org/get/)
3. 然后将[plugins-manager.jar](https://jmeter-plugins.org/get/)放到Jmeter中的目录lib/ext下，重启Jmeter
4. 可以看到“选项”下多了个“plugins manager”，如下图



1. 该插件能管理Jmeter插件，下载升级，如果要下载新插件，只需在“Available Plugins”中勾选，然后点击“Apply Changes and Restart Jmeter”，等待插件下载完，自动重启Jmeter即可。

