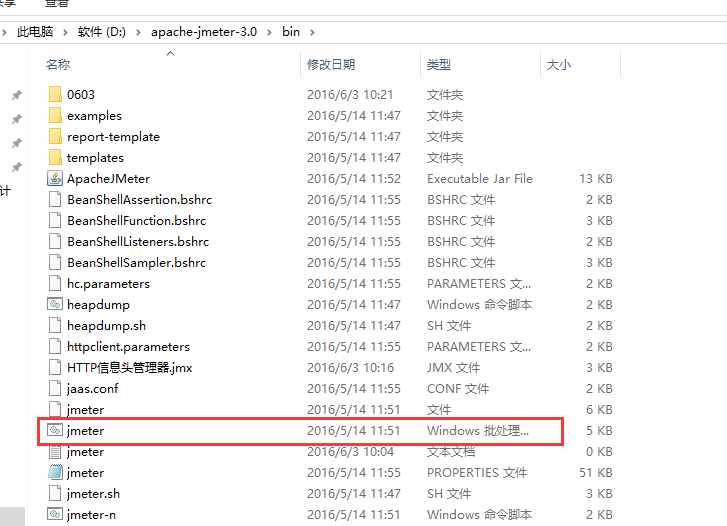
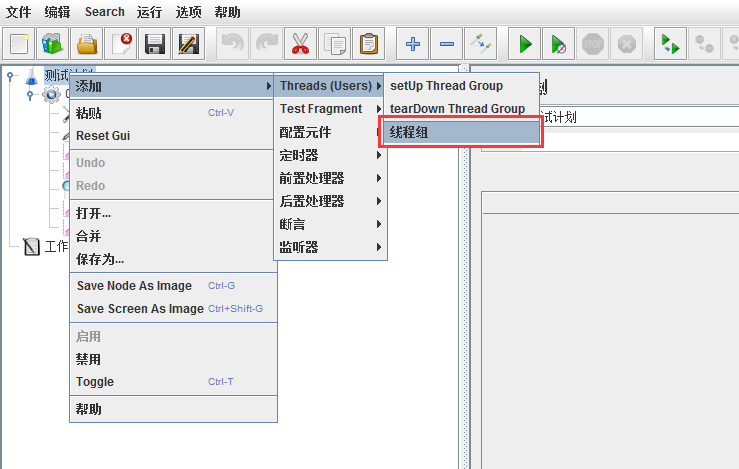
1. 下载jmeter。



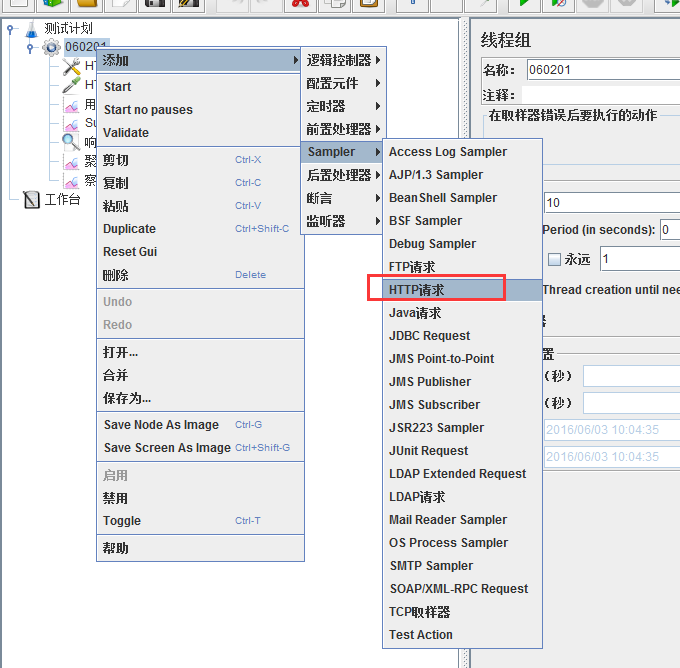
1. 运行bin文件夹下的jmeter批处理文件。



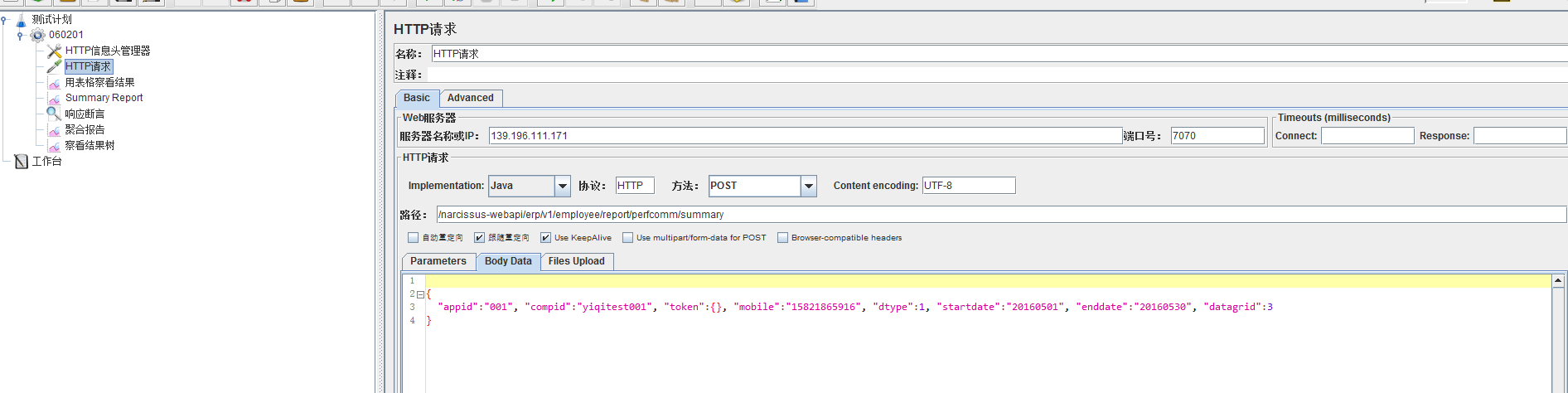
1. 右键测试计划-添加线程组。



1. 右击线程组，添加http请求。



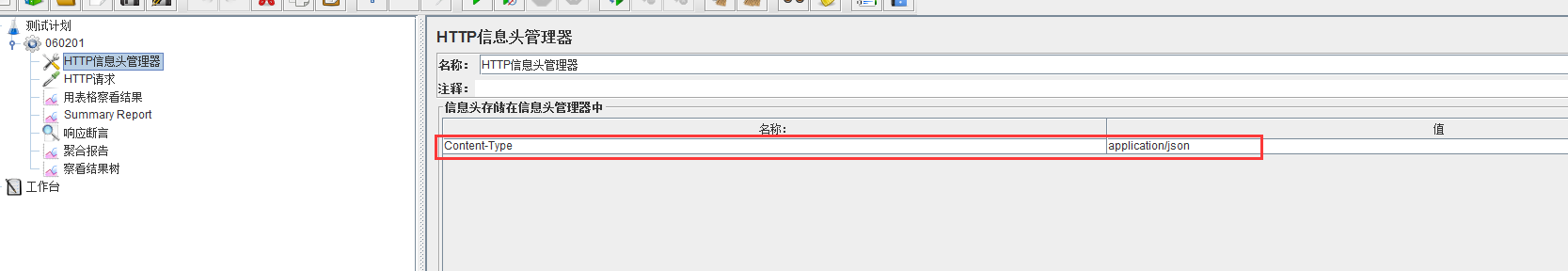
1. http请求配置如下。Content encoding(utf-8) 编码



1. 添加http信息头管理器。



1. 信息头管理器配置。



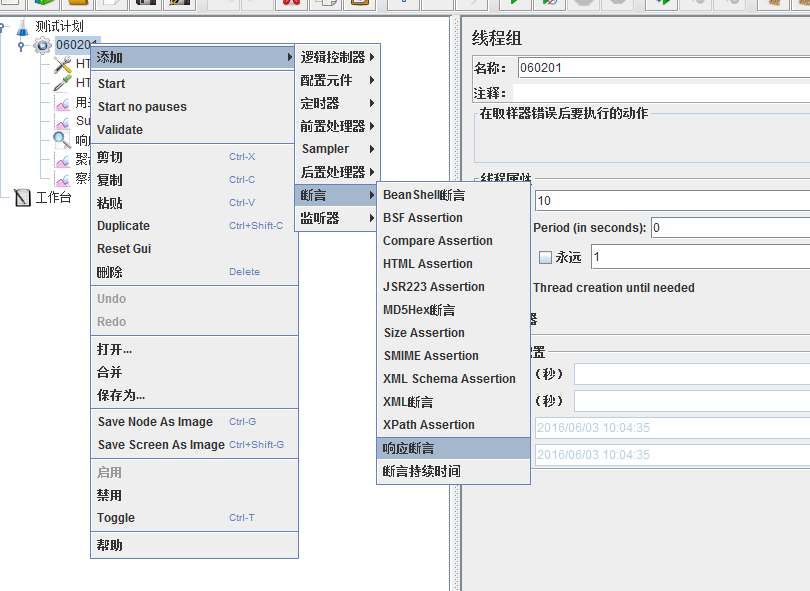
1.Accept属于请求头， Content-Type属于实体头。   
Http报头分为通用报头，请求报头，响应报头和实体报头。   
请求方的http报头结构：通用报头|请求报头|实体报头   
响应方的http报头结构：通用报头|响应报头|实体报头

2.Accept代表发送端（客户端）希望接受的数据类型。   
比如：Accept：text/xml;   
代表客户端希望接受的数据类型是xml类型

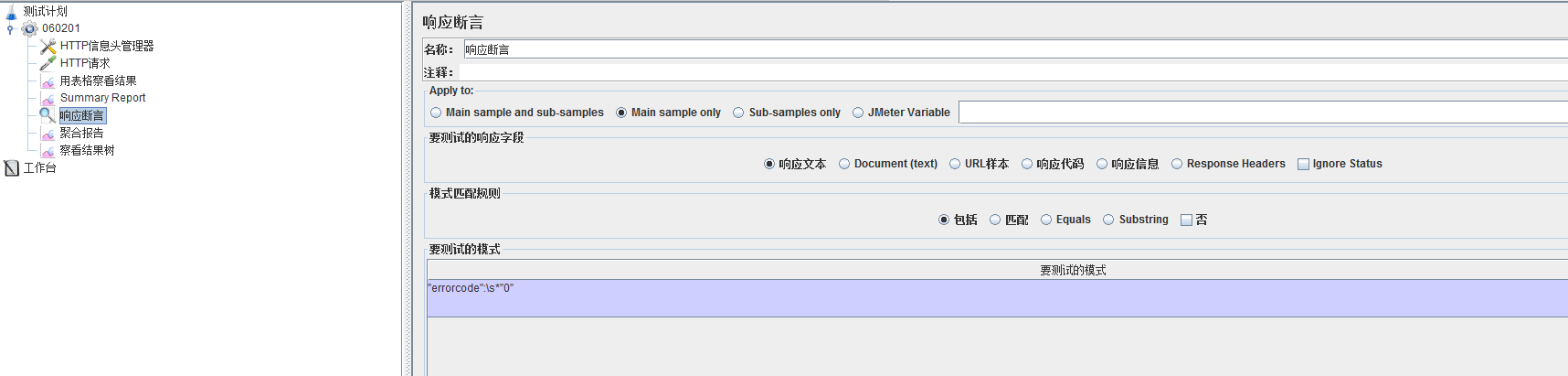
Content-Type代表发送端（客户端|服务器）发送的实体数据的数据类型。   
比如：Content-Type：text/html;   
代表发送端发送的数据格式是html。

二者合起来，   
Accept:text/xml；   
Content-Type:text/html   
即代表希望接受的数据类型是xml格式，本次请求发送的数据格式是html

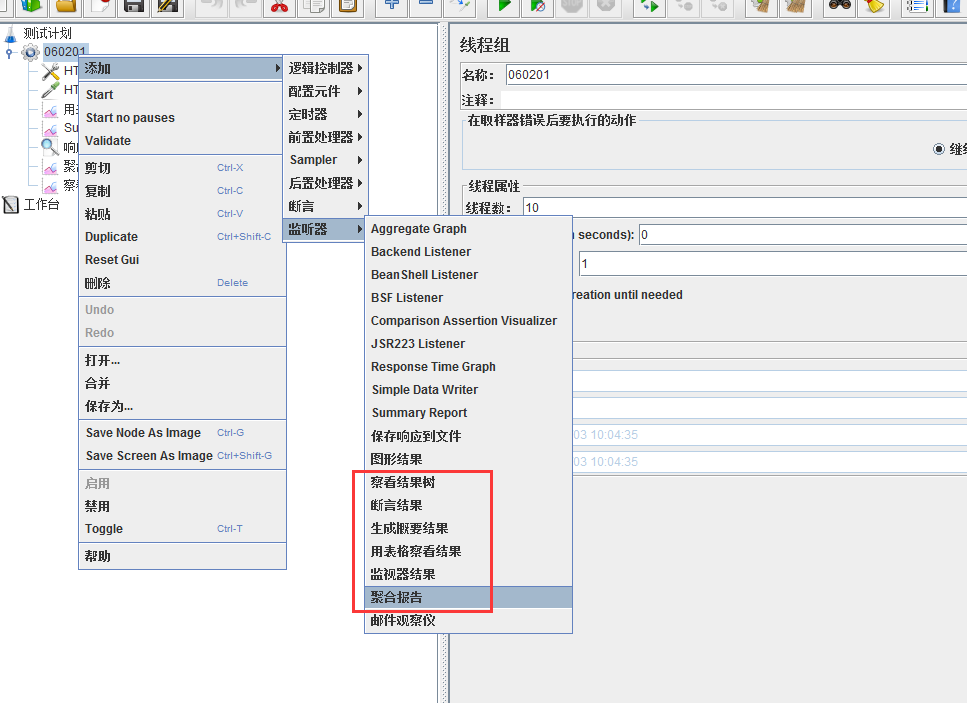
1. 添加响应断言-响应断言主要是为了判断响应结果是不是对应执行结果，如果不加，error的判断有误，无法真正看到运行结果。



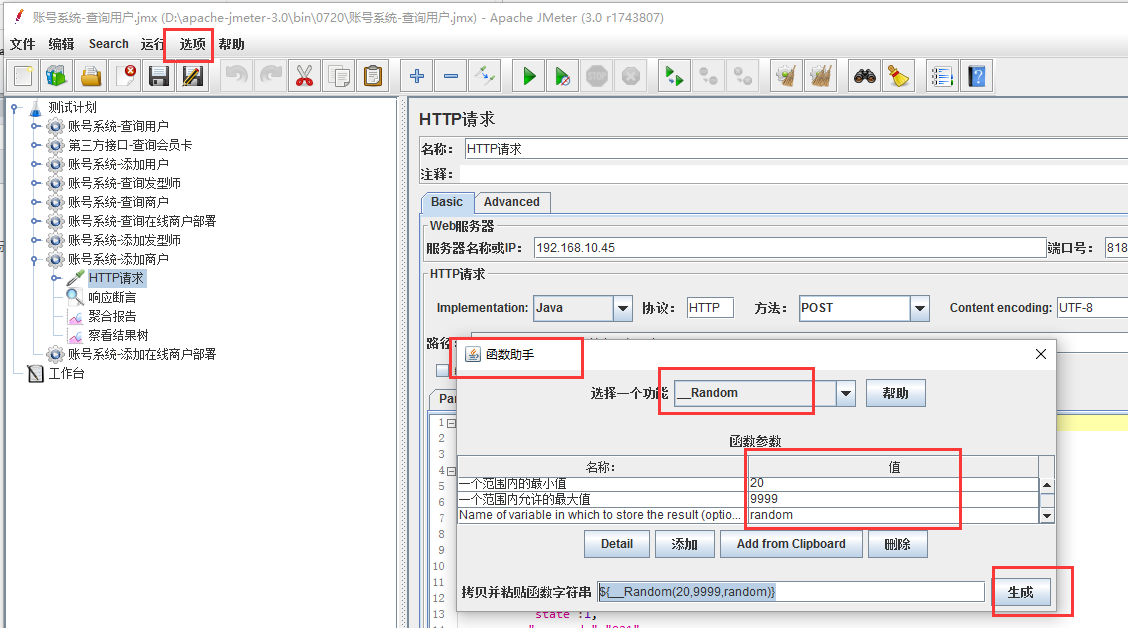
1. 响应断言配置。



1. 添加监听-各种结果。



1. 函数使用：使用选项中的函数助手，选择要使用的函数功能



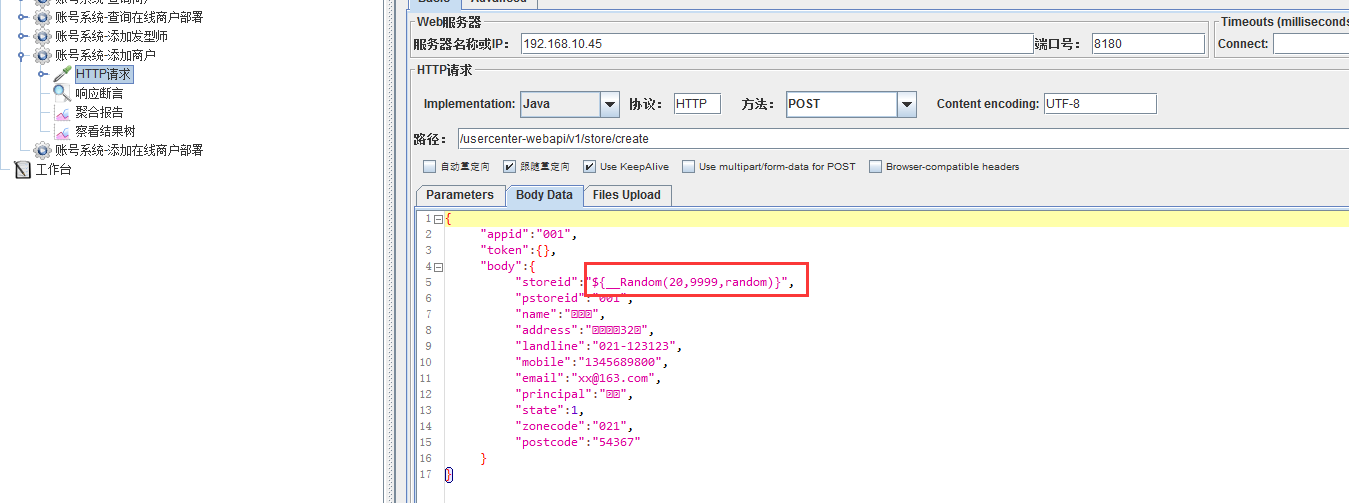
一个范围内的最小值：即我们所要取的随机数的最小值，上述设置为1，生成的随机数将不会小于1

一个范围内允许的最大值：即我们所要取的随机数的最大值，上述设置为100，生成的随机数将不能超过100

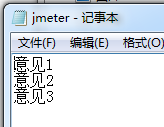
Name of variable in which to store the result(optional)为函数名称名称：这里我们设置为random，即用于存储在测试计划中其他的方式使用的值

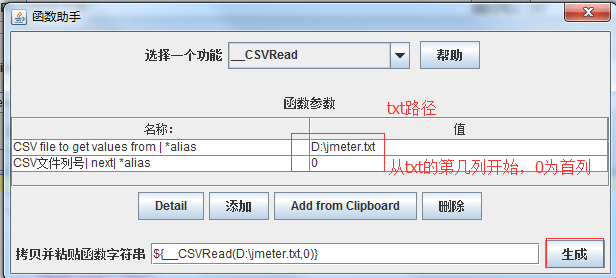
点击【生成】按钮即可得到我们所需要的函数：${\_\_Random(1,100,random)}，然后将函数复制到需要用到随机数的地方，我们就可以使用啦。

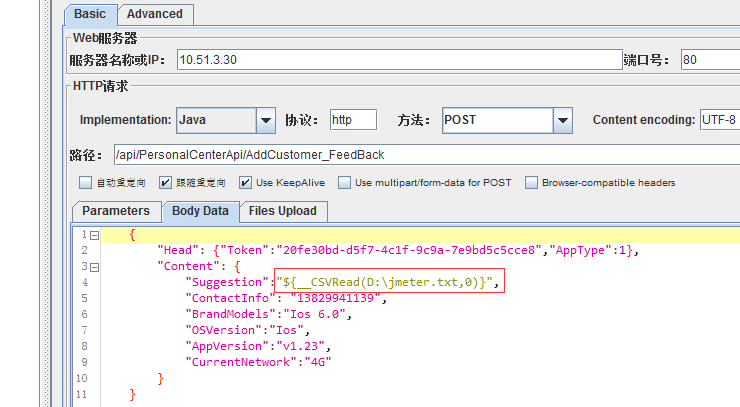
注意：当我们设置的线程数超过随机数范围时，随机数将会重复生成



函数\_CSVRead

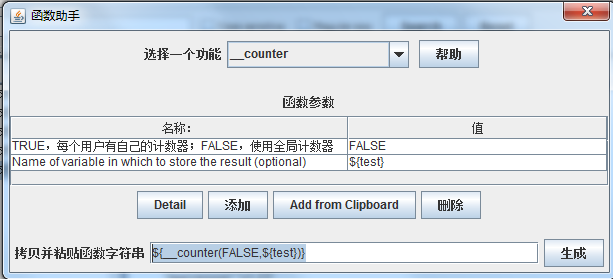


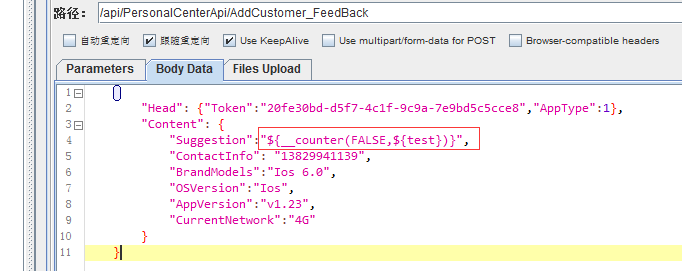


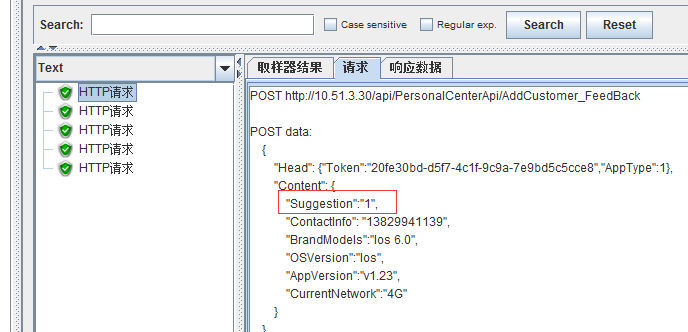


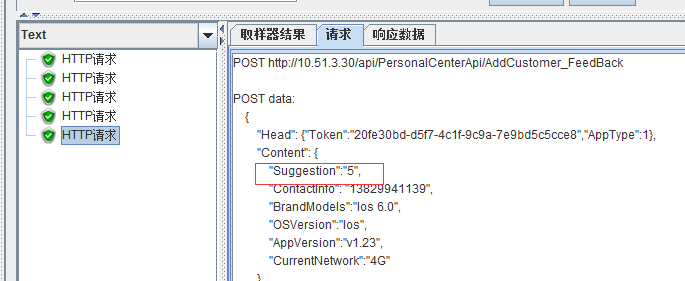
其中的值即为我们的生成的函数(${\_\_CSVRead(E:\c.txt,0)})，jmeter执行时，如果是多线程，则从c.txt中第一行开始读取，如果设置的线程数大于c.txt中的行数，将会循环读取数据，通常该行数可用于参数化

函数\_counter（计数器）









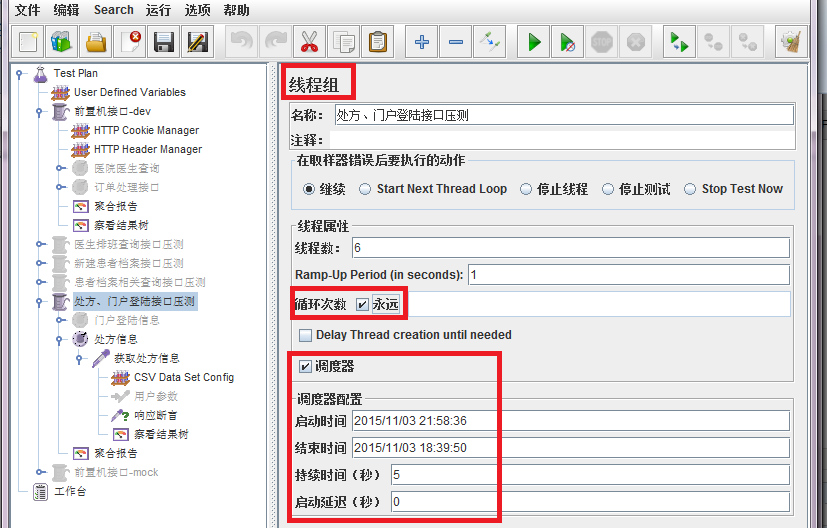
关于函数学习：

http://www.tuicool.com/articles/fq2qMb

1. 调度器配置：

Jmeter的线程组设置里有一个调配器设置，用于设置该线程组下脚本执行的开始时间、结束时间、持续时间及启动延迟时间。当需要半夜执行性能测试时会用到这个功能。

ps:设置调度器配置，需要将前面的循环次数设置为永远。 注意：如果我们需要用到调度器来设定持续时间，如果线程数不够多到持续时间结束，我们就必须将循环次数勾选为永远，特别地，如果线程组里面有其他的循环，我们也需将该循环次数勾选为永远（如我上面录制的脚本中的Step1也是一个循环，需要将永远勾选），否则，按我如上配置，将永远去掉勾选，文本里填1，那么无论你将持续时间启动时间结束时间等设置多少，系统运行1000次后，将会停止不再运行



启动时间:测试计划什么时候启动，启动延迟会覆盖它。当启动时间已过，手动运行脚本时也当前时间也会覆盖它(但启动时间页面显示不会变)。

结束时间：测试计划什么时候结束，持续时间会覆盖它。

持续时间（秒）：测试计划持续多长时间，会覆盖结束时间。

启动延迟（秒）：测试计划延迟多长时间启动，会覆盖启动时间。

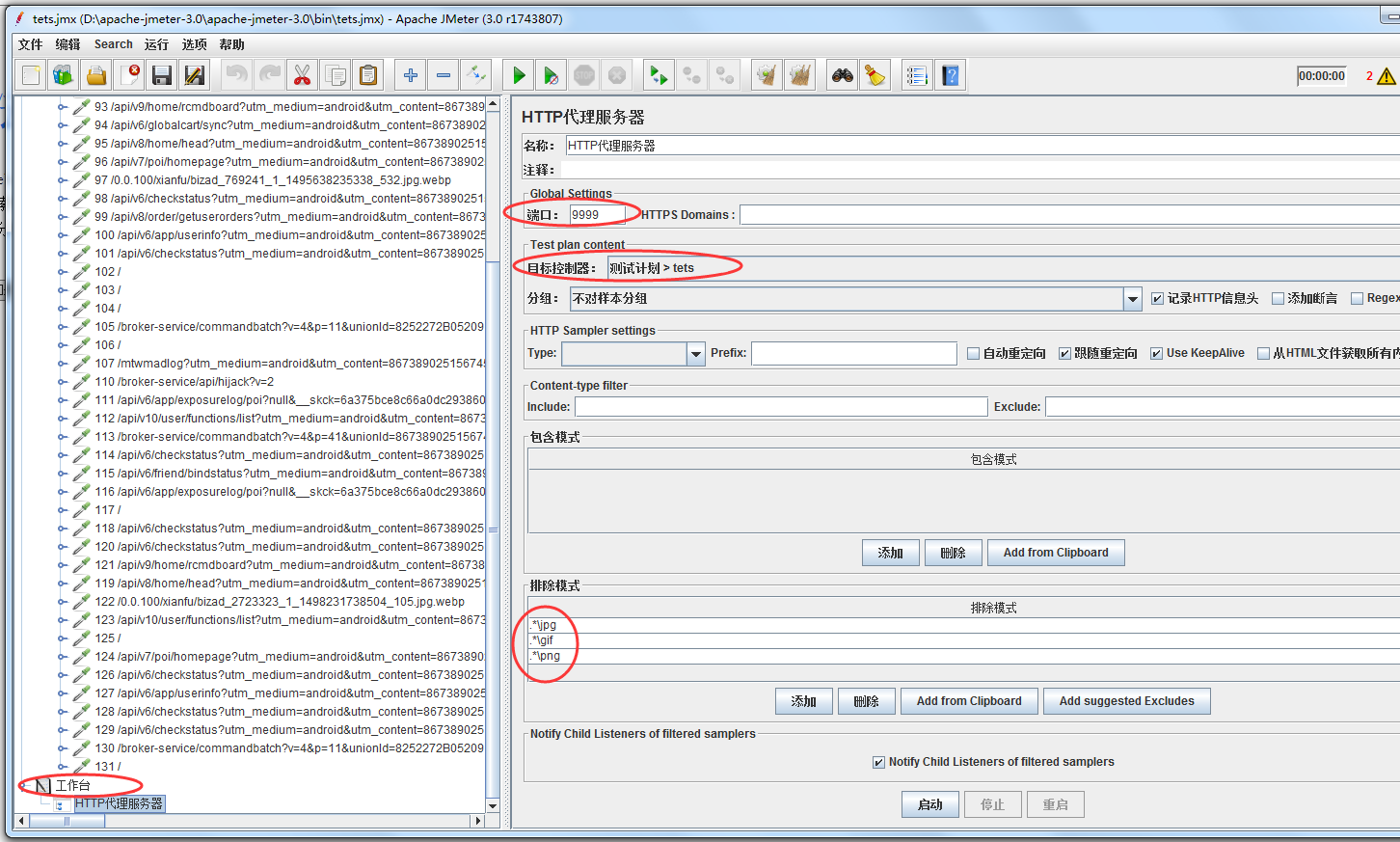
工作台右键-http代理服务器，设置端口（任意值）和目标控制器（测试计划>XXXX(线程组名称)），排除模式是录制到的脚本中含有设置的排除模式内容，则不保存下来。

例如 .\*\png .\*\jpg .\*\gif 图片就不展示了

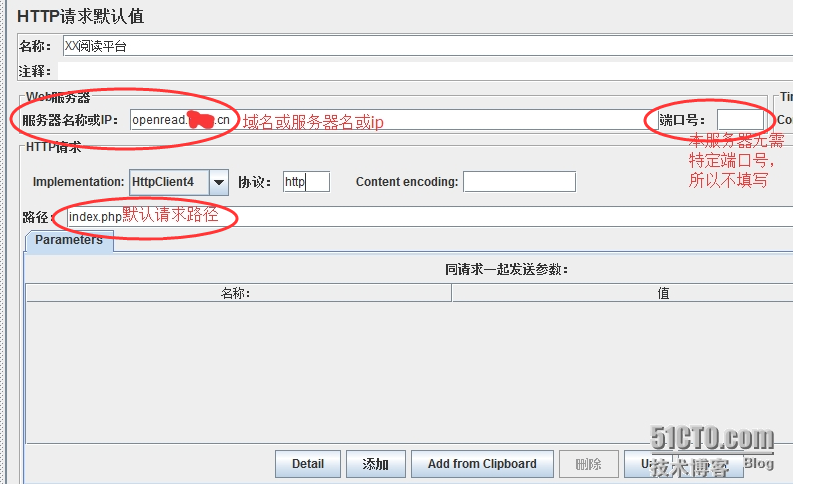
包含模式例如：

.\*\.html

设置完代理服务器，点启动才可以用，手机配置代理（地址为本机IP，端口为设置端口），打开app，例如登录-退出。然后停止服务，所录制的脚本在所保存的位置下都可以看到。



http请求默认值：该组件可以为我们的http请求设置默认的值。假如，我们创建一个测试计划有很多个请求且都是发送到相同的server，这时我们只需添加一个 Http request defaults组件并设置“Server Name or IP”，然后添加多个http请求且不设置"server name or ip"，这些http请求会默认使用Http request defaults组件设置的值。

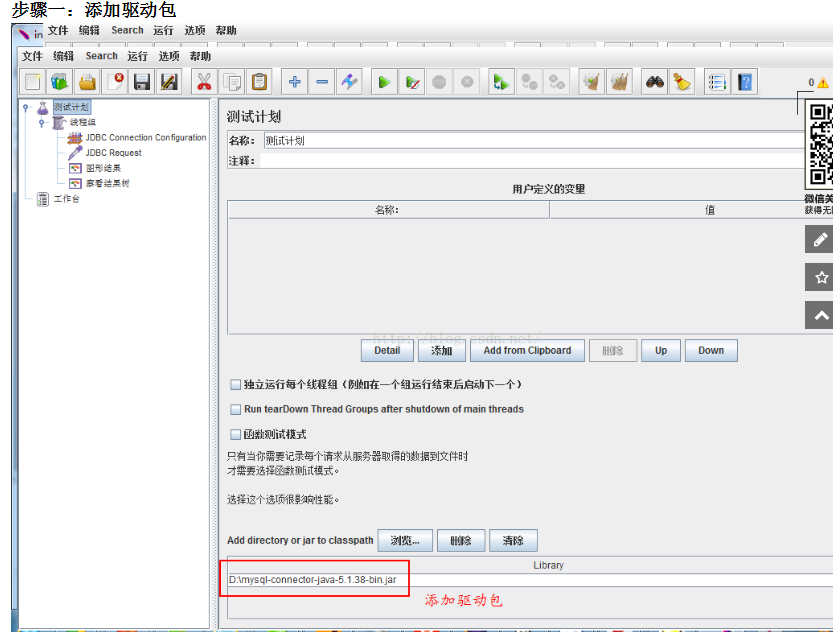


JDBC

数据库性能测试实例：

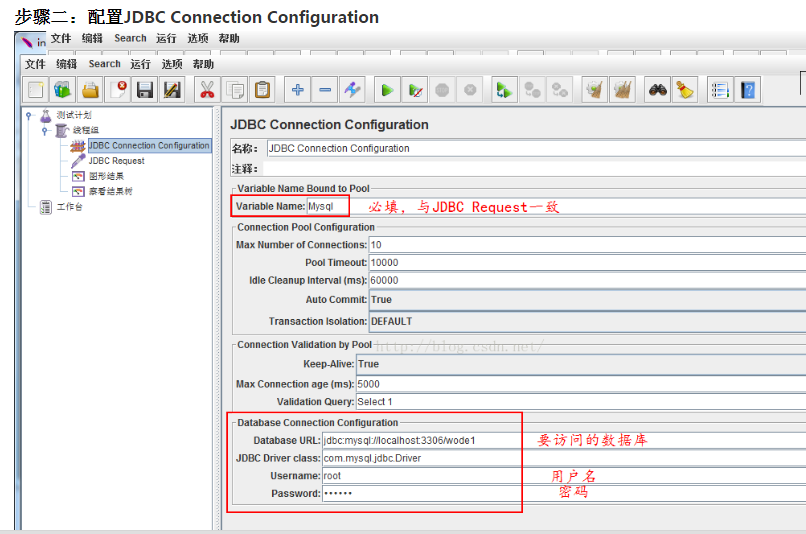
通过如下图方式或者如下将jar包放在lib目录下的方式均可以。

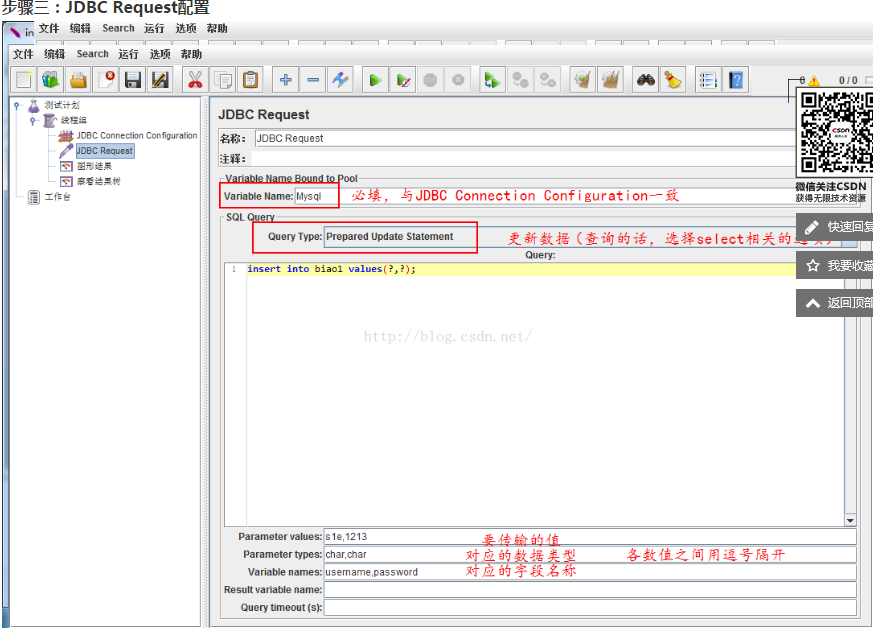
* mysql数据库：无需引入其他数据库驱动jar包。
* sql server 数据库：下载sqljdbc4.jar 放到 jmeter根目录的lib目录下
* oracle数据库：将oracle数据的安装目录下面的\product\10.2.0\db\_1\jdbc\lib\ojdbc14.jar 放到jmeter根目录下的lib目录下



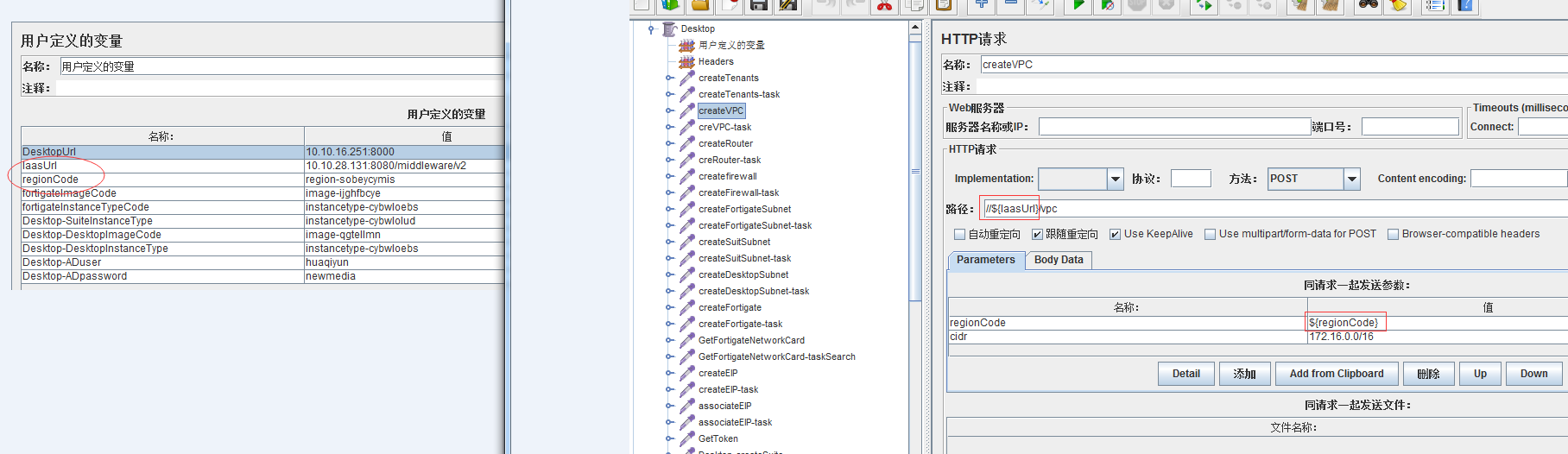
Microsoft+SQL+Server+JDBC+Driver+3.0sqljdbc\_3.0.1301.101\_chs.exe

--下载并解压把sqljdbc4.jar中的放在apache-jmeter-2.6\lib下面





用户自定义变量：当前的线程组内所有取样器（即博主的HTTP请求）都可以引用变量



需要说明的是，服务器IP地址和端口号以及接口共同的请求路径部分，作为变量引用时，需要在路径填充表格的最前面添加两个斜杠“//”，不加的话会引用失败。

传递的参数值是变量，使用${变量名}的方式

正则表达式：

| **参数** | **释义** |
| --- | --- |
| 引用名称 | 在HTTP等请求中，引用此数据，需要用到的名称 |
| 正则表达式 | 用于将需要的数据提取出来 |
| 模板 | 表示使用提取到的第几个值： $-1$:表示取所有值 $0$:表示随机取值 $1$:表示取第1个 $2$:表示取第二个 以此类推:$n$:表示取第n个 |
| 匹配数字（0代表随机） | 0 代表随机取值，1 代表全部取值 |
| 缺省值 | 如果正则表达式没有搜找到值，则使用此缺省值 |

()：封装了待返回的匹配字符串。

.：匹配任何字符串。

+：一次或多次。

?：在找到第一个匹配项后停止

http://www.jianshu.com/p/c49e2dfd109b