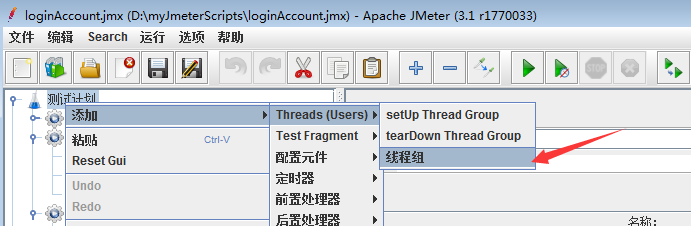
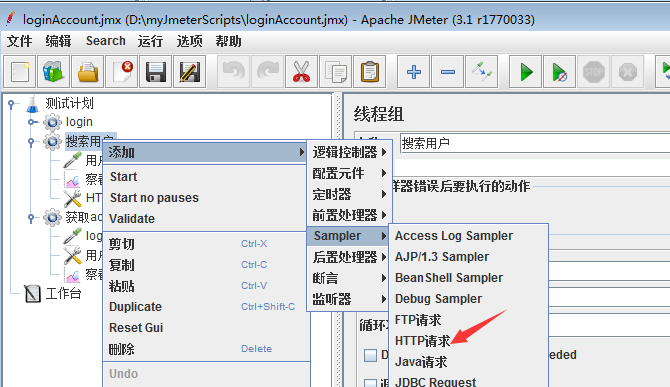
Jmeter接口测试

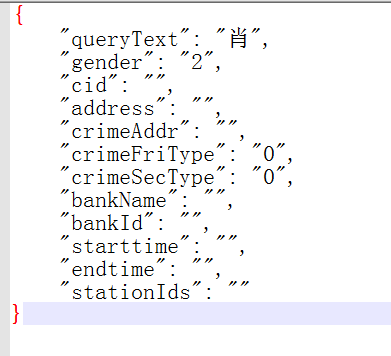
1. 新建线程组

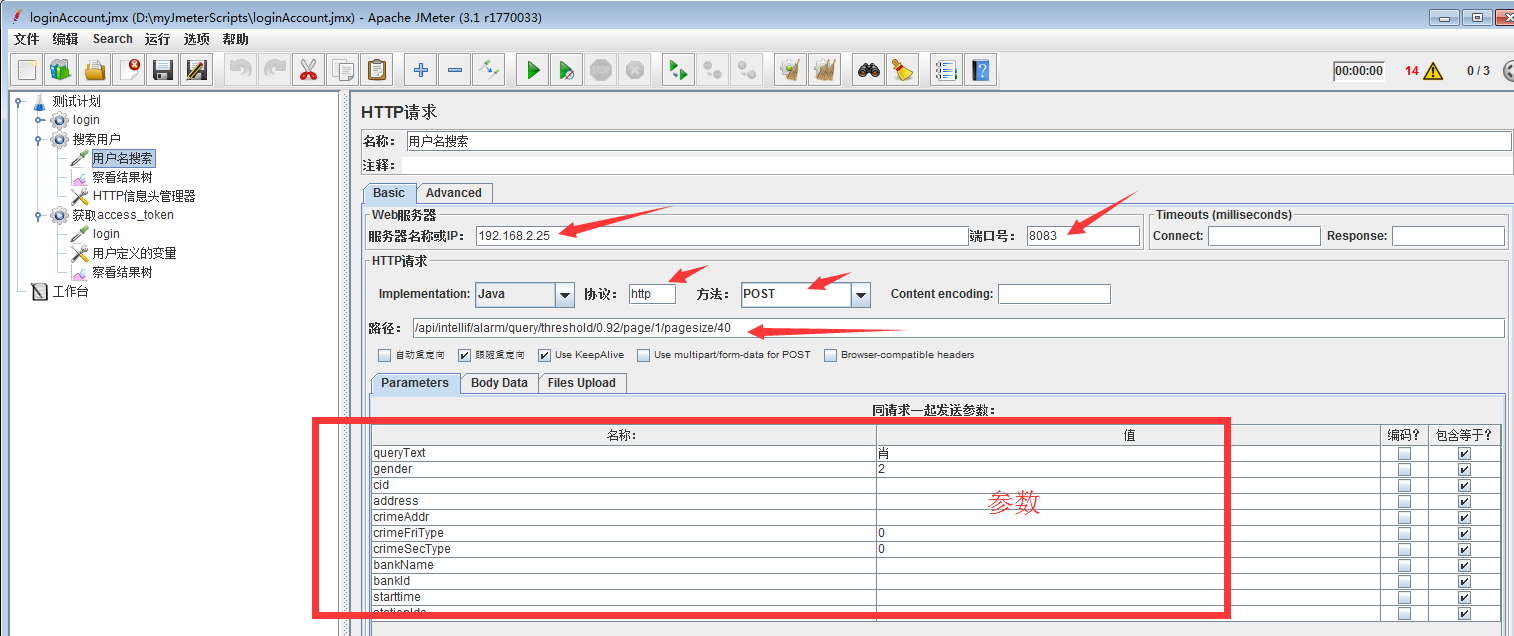


1. 在线程组上面添加HTTP请求

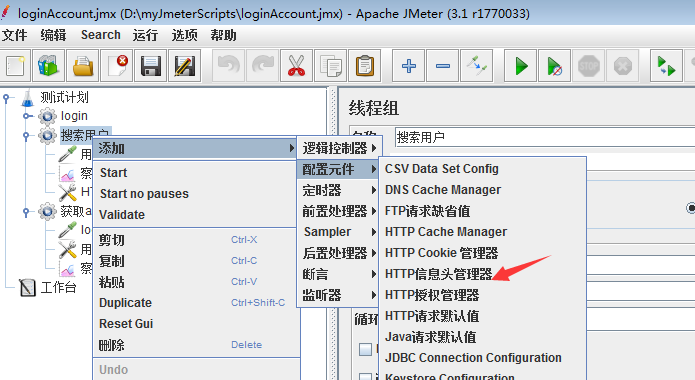


设置HTTP请求参数，即接口传参，根据接口文档来获取（这里的参数是可以表格化的，编写测试用例，然后导入依次执行—目前我还不会）

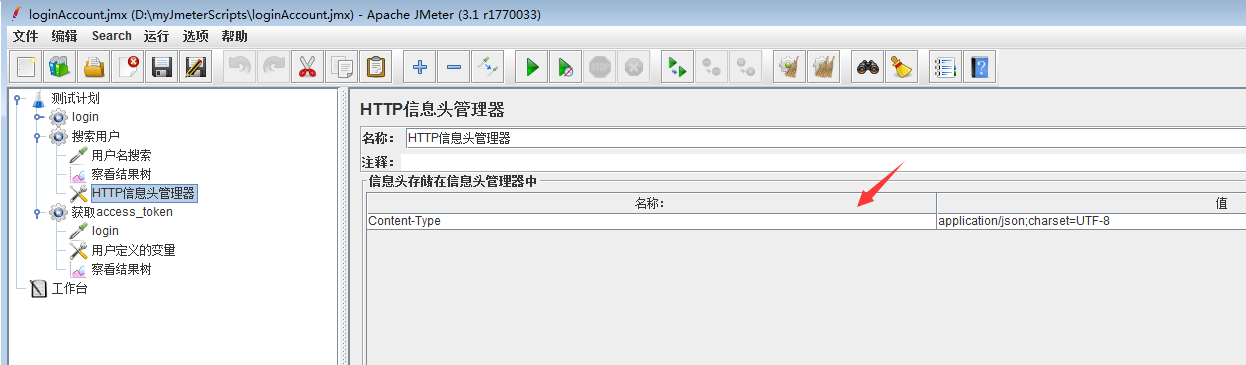




1. 添加HTTP信息头管理器

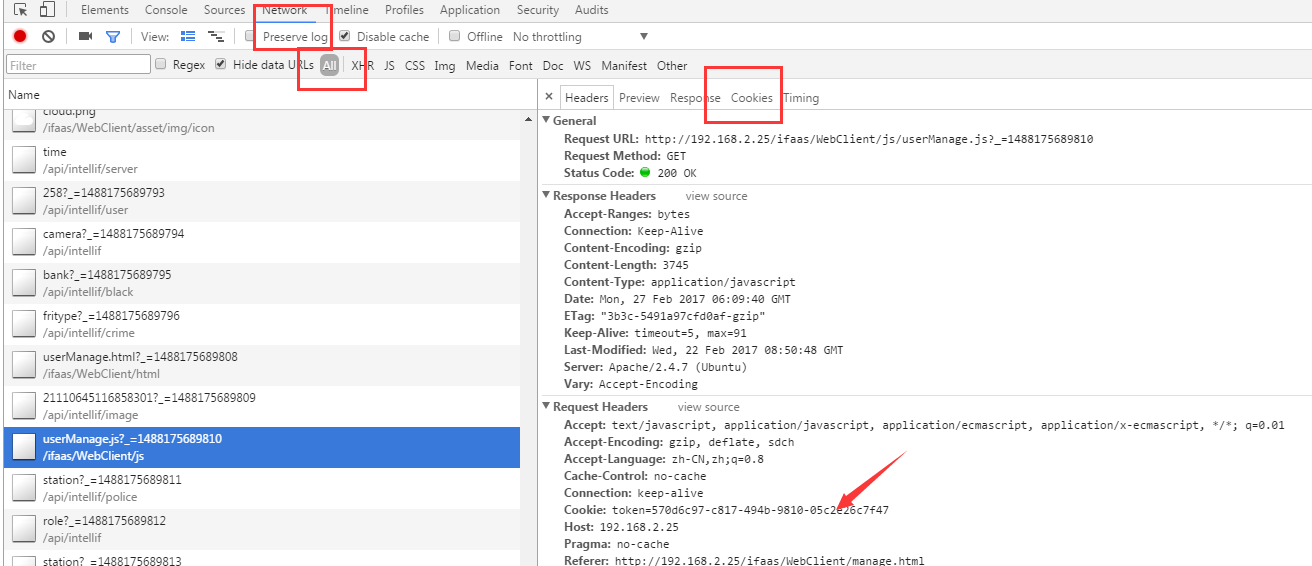


设置HTTP消息头的参数（例如这里设置为：Content-Type，application/json;charset=UTF-8）



（2）添加HTTP Cookies信息

在部分接口的请求头当中，需要有cookies信息，一般会将cookies信息添加到http信息头当中，cookies，此处就相当于在http信息头当中添加的认证信息token值，会判断这个值是否有效，继而是否打开页面。（后续再有其他应用，再进行扩展）

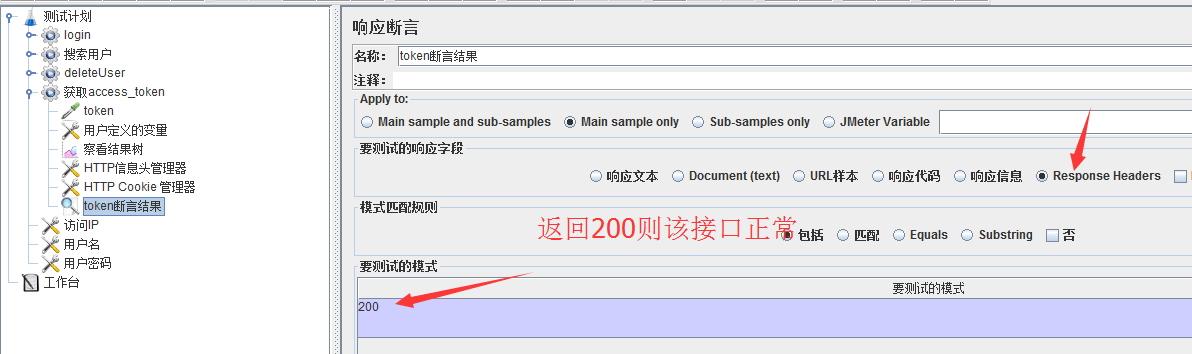


1. 添加断言（需要扩充）

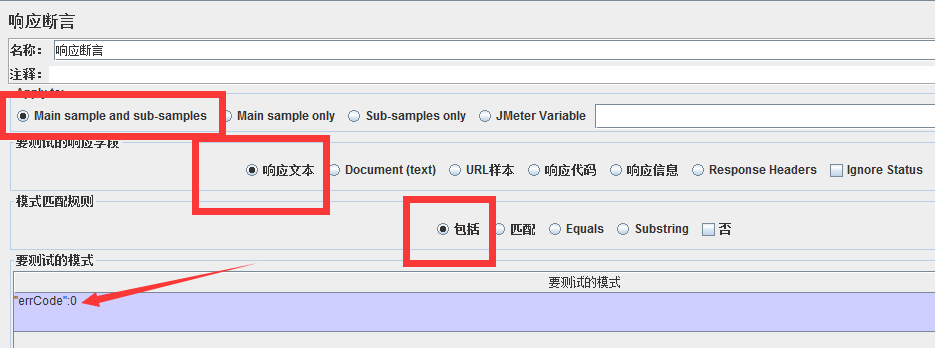
http://www.51testing.com/html/12/252512-223704.html

1. 添加断言：添加-断言-响应断言
2. 设置断言

可以设置，检查http返回码来来查询结果，例如：

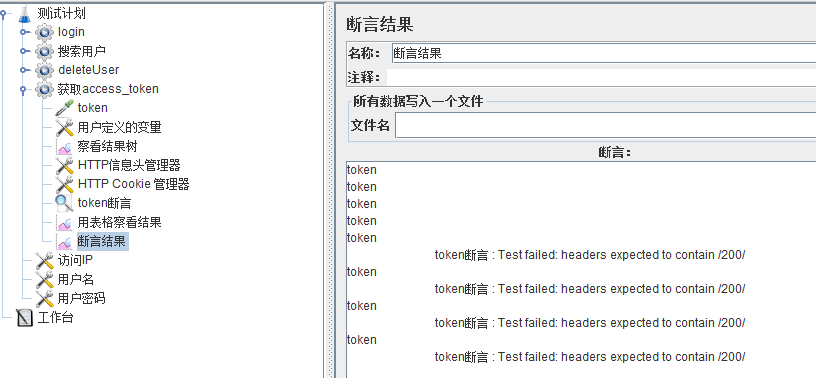


匹配response body当中返回的内容：



1. 查看断言结果：从添加-监听器-断言结果，添加断言结果查看

查看结果：



补充：我应该如何知道哪些传参不通过，不通过的原因是什么？

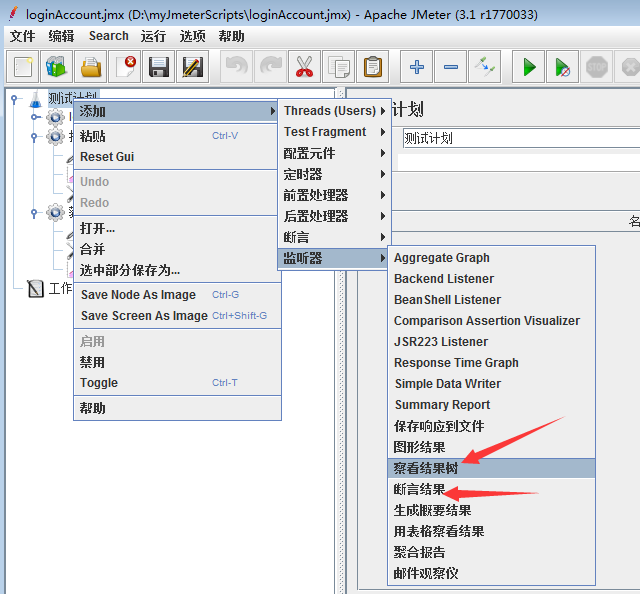
1. 如何获取当前线程

目的：我每次从数据库当中读取第一条数据，如果操作失败，这条数据仍然会在，不做相关处理，我永远只能读取到这条数据，现在我需要做处理，根据当前线程的编号，来获取读取数据当中的第几个变量。

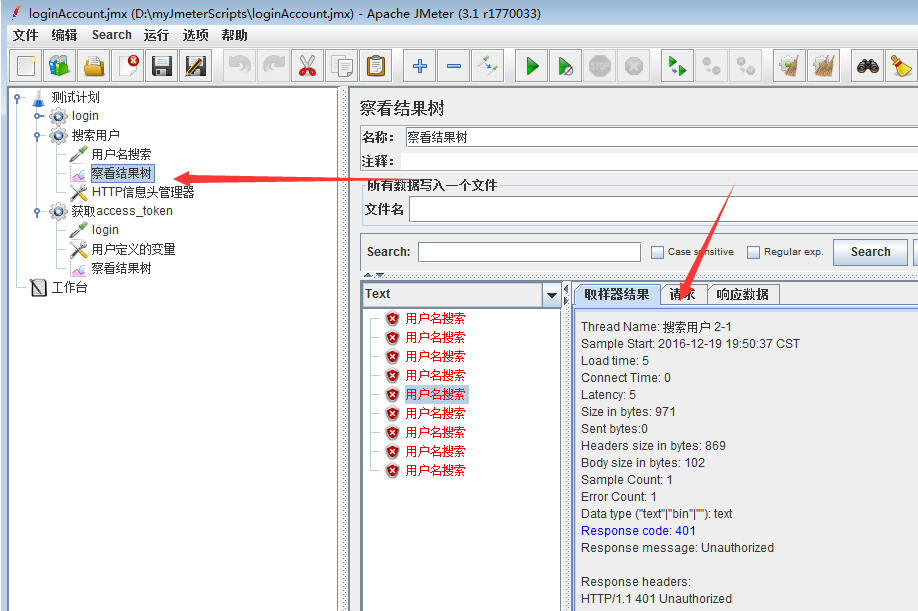
另外一种方式：将数据库数据存储为一个变量，然后依次获取其内的值，赋值给当前运行的http请求。

第三种方法：断言失败，我即终止运行

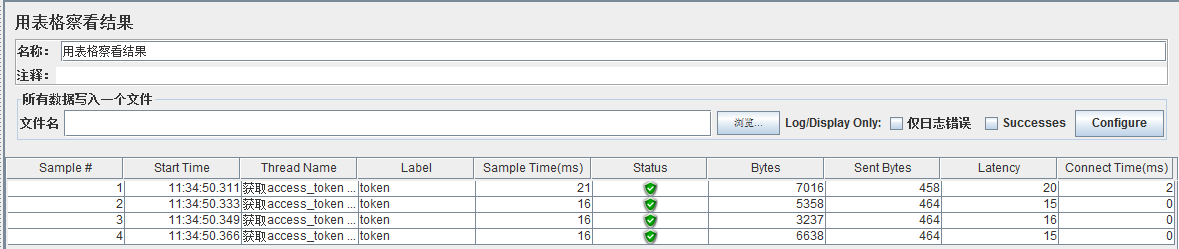
1. 查看运行结果



（1）在结果树中查看结果，还有另外几种查看结果的方式不熟悉。



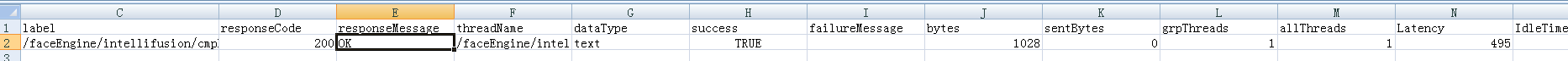
（2）表格查看结果



（3）查看图形结果

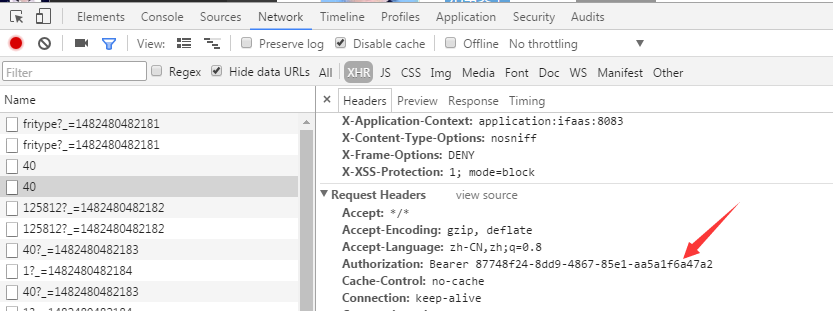
可以用图形的方式查看结果，并且将结果保存在csv文件当中，csv会记录线程的各项信息，方便查看。



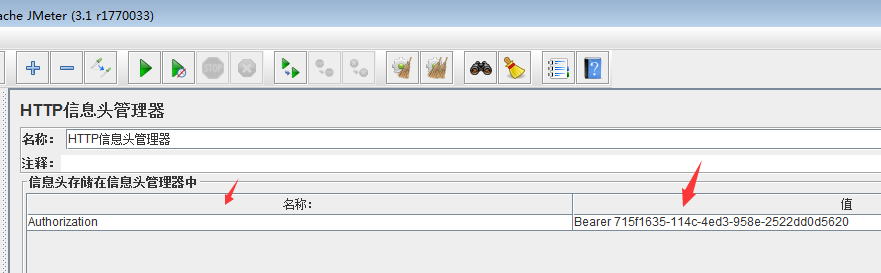


6、身份认证

（1）用户每次登录系统后，都会有一个身份，如图所示：



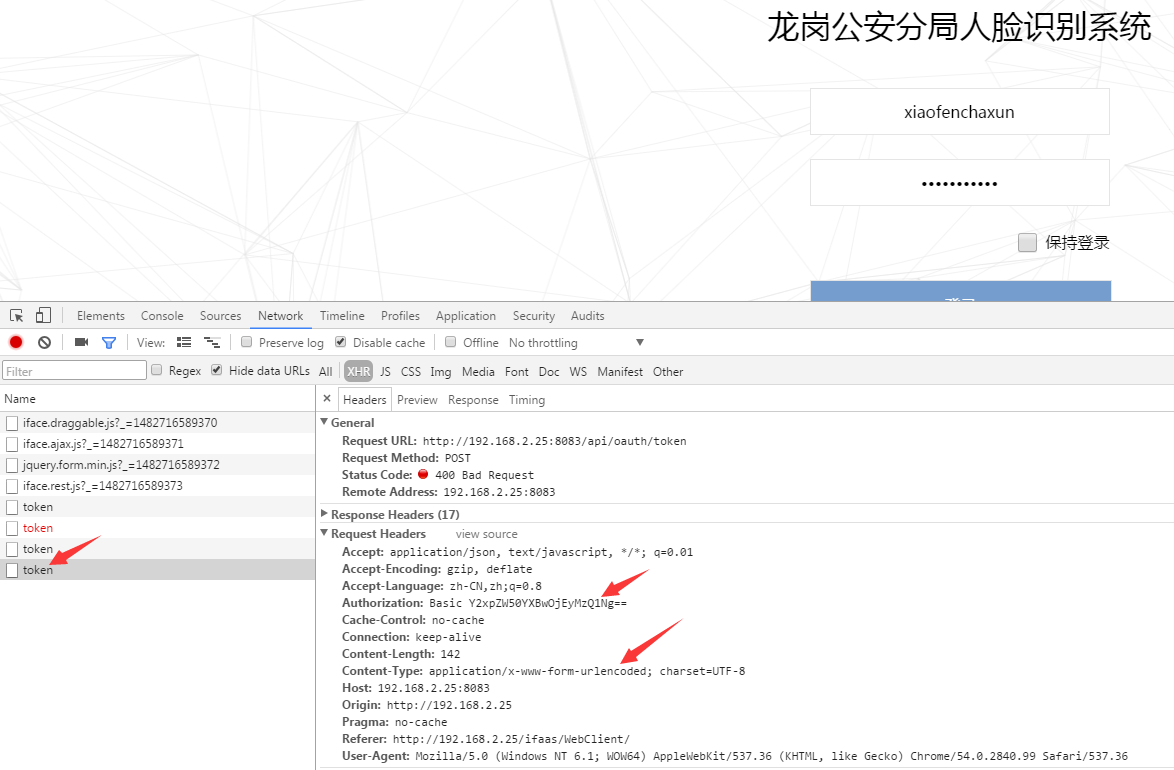
这个字符串用户标注用户可以进行操作，在进行接口调用时，需要将这个身份认证信息传递给接口，添加进HTTP请求头信息当中：（这个身份信息每次重新登录都会进行刷新）



（2）如何获取该信息头

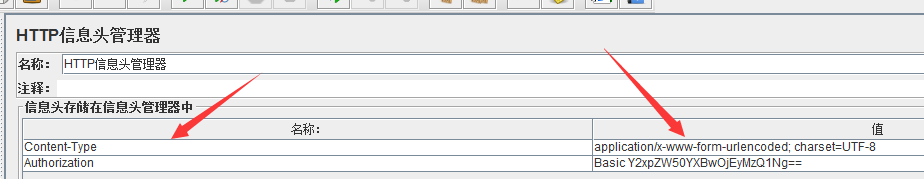
Web api中有1个接口，/api/oauth/token，是用于身份验证的一个接口。他返回给用户一个token，通过这个token，用户才可以正常的访问系统。

通过F12在web登录界面，输入用户名和密码（例如输入错误密码），即可看到调用该接口（问题：为什么正常登录，在历史记录中看不到调用该接口）。如下图所示：



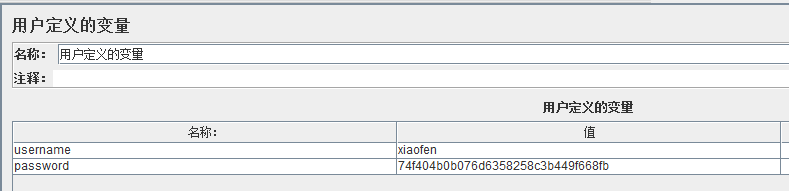
Authorization:Basic Y2xpZW50YXBwOjEyMzQ1Ng==和

Content-Type:application/x-www-form-urlencoded; charset=UTF-8都是HTTP信息头里面的内容，需要将其添加进调用接口的HTTP信息头里面，如下图所示：



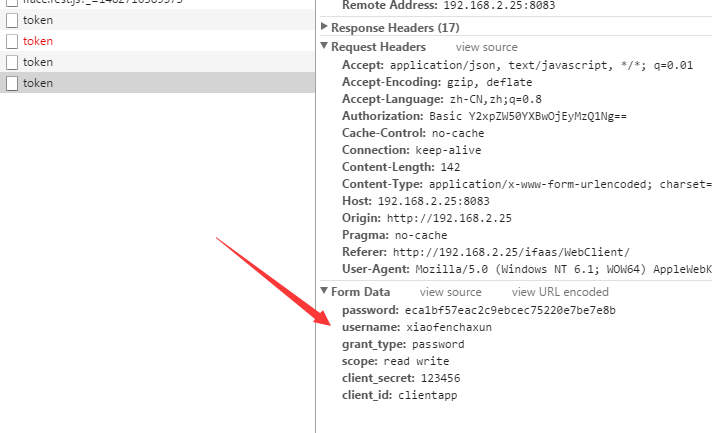
（3）token接口

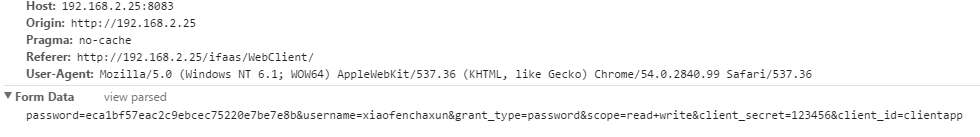
添加token接口，如上（2）所述，需要调用token接口，token接口的代用需要用户名和密码以及其它信息，用户名和密码可以进行封装，以用户变量的形式给出，方便后续的维护，如下图所示：添加-配置元件-添加用户变量



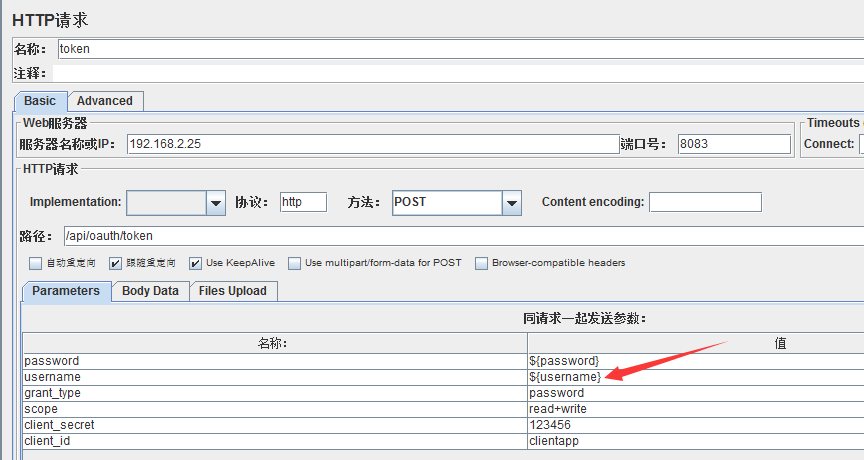
Token请求参数整理：

F12查看http请求参数：



****

Jmeter当中的传参：



7、如何将6中身份认证获取的token，传给所有当前测试的接口

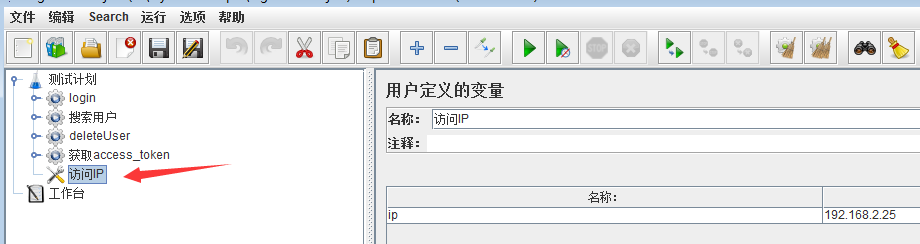
8、Jmeter参数化的3种方法

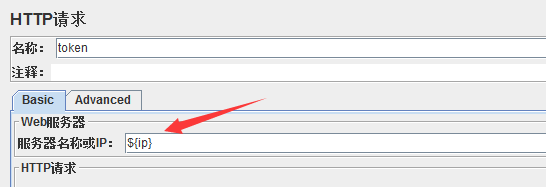
Jmeter变量大小写敏感

（1）用户定义的变量：这种就是为了方便管理参数，只能有一个值，比如说ip地址不经常变化的。

value就可以了，key就是这个参数的名称，也就是你在脚本里面取的值，value就是具体值了。在取参数化的值的时候，使用${name}这样去取值，name就是你取的变量名称。

注：所有地方的变量引用都是用：${username}方式进行引用，包括在bodydata当中





（2）函数生成器：函数生成器可以参照一定的规则生成数据，这样的比如说生成一些随机数

函数助手的话，可以按照规则生成一些参数，比如说随机数取当前时间，最常用的就是这两种。

举例：在某些指定字符中进行选择，随机生成一个用户名

例如生成字符串：RandomString

设置读取文件：CSVRead（教程地址：<http://blog.sina.com.cn/s/blog_9c5151b00102vui0.html>)

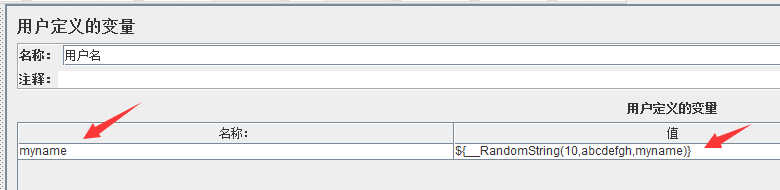
创建一个函数：



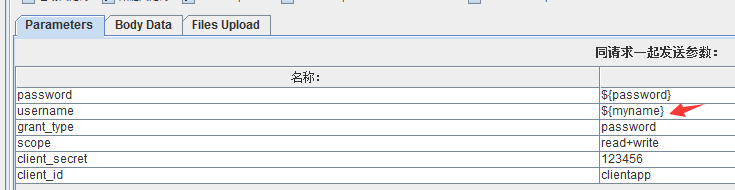
设置函数规则，按规则生成变量/值：



定义用户变量：



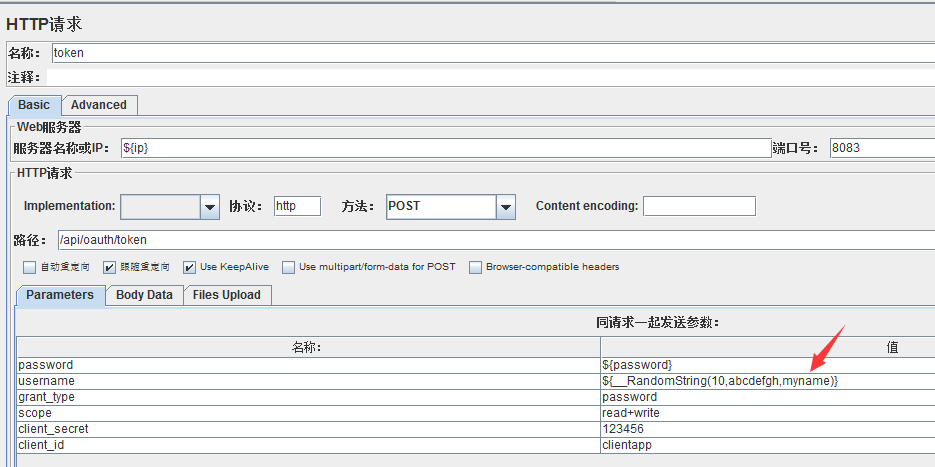
接口调用中，使用该变量：



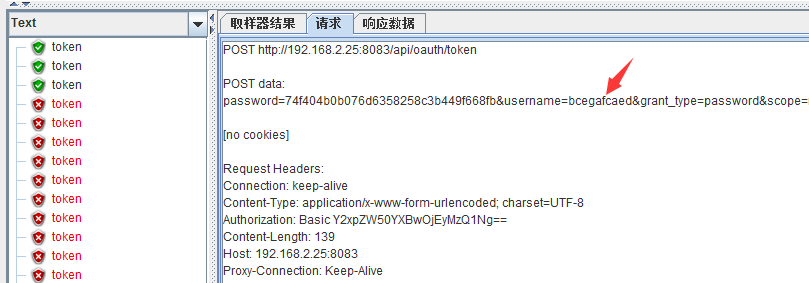
注：参数化时，需要将特殊的符号，如”和/等进行转义，变成：\”和\\。

第二种方法：

直接调用：



运行结果：



（3）从文件中读取：文件读取就是事先写好一些数据，然后从文件中读取，这样的话，比如说登录接口，账号和密码都是我们事先注册好的。

从文件中读取的话，三个步骤

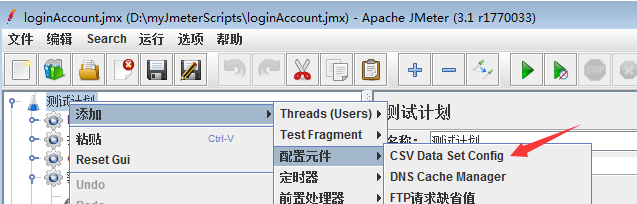
        1)、读取文件

        2)、取文件内容里面的参数，给它一个名字

        3)、使用值

从文件读取的话，需要在线程组里面添加一个CSV Data Set Config，它就是做前面两步的操作的。

添加CSV Data Set Config，添加-配置元件-CSV Data Set Config。

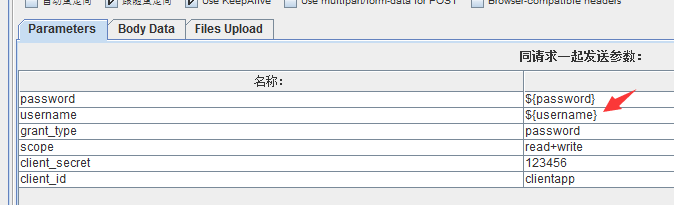


举例：将用户名和密码分别写入文件，然后分别依次读取用户名和密码，进行登录。

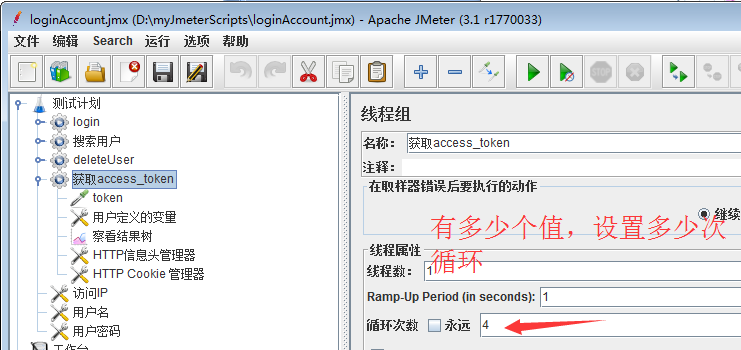
Jmeter文件读取集：



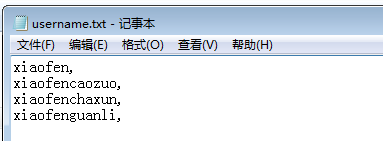
使用该变量集：



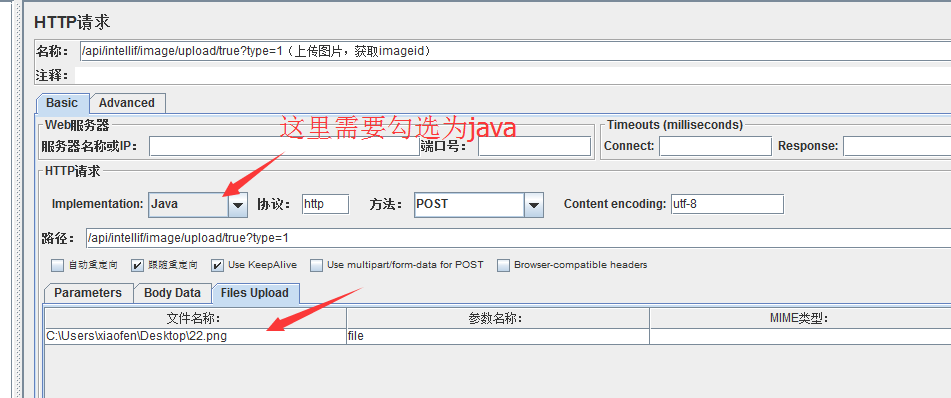
根据变量数，设置线程循环次数：



Txt文件数据：



## （4）上传附件文件参数化（files upload）



1. 跨线程传递参数

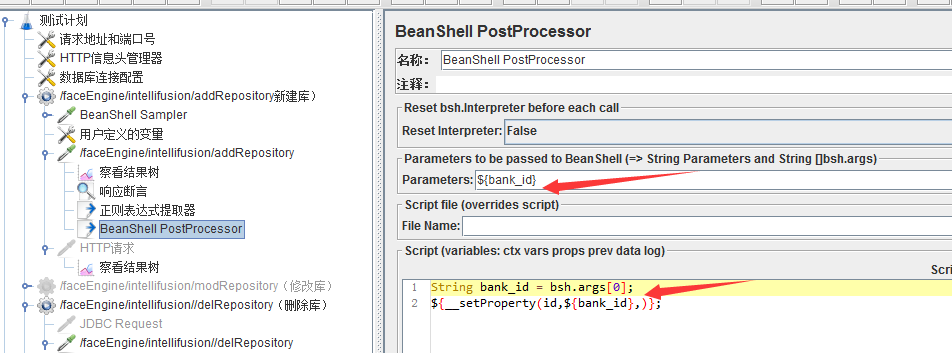
举例：将上一个线程组的返回结果当中的bankid传递给下一个线程组

1. 用正则表达式，将需要的数据data1从线程组A的response中提取出来；
2. 在线程组A下新建：Beanshell postProcessor，将data1赋值给全局变量data2；

赋值的语句为：

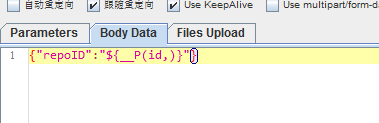
String bank\_id = bsh.args[0];

${\_\_setProperty(id,${bank\_id},)};



1. 在线程组B当中引用全局变量data2；

引用的格式为：${\_\_P(id,)}，如下所示：

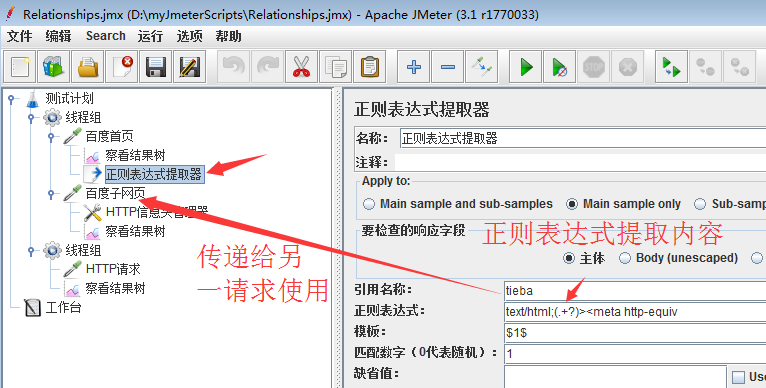


# 9、Jmeter关联（此处的具体实现还存在问题）

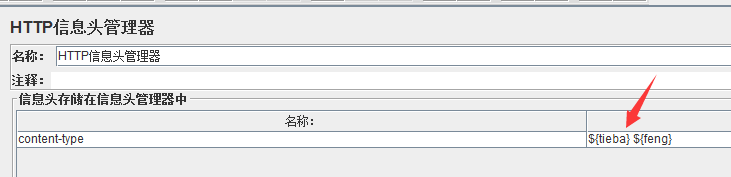
关联的作用：同线程组之间的HTTP请求，互相传递返回值给另外的接口作为参数，或进行一些其他的处理，以进行整个业务流程的走通。（跨线程组之间目前我的操作经验是无法成功的，后续可以再找找方法）

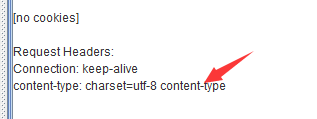
实现方法：Jmeter中的关联是通过正则表达式提取器来完成的。

（1）获取方法：选择某1个http请求，添加-后处理器-正则表达式提取器



（2）可以定义多个提取器，然后需要使用时，直接将所提取的变量进行拼接使用即可。如下：





正则表达式匹配：

a. 引用名称：下一个请求要引用的参数名称，如填写title，则可用${title}引用它。

b. 正则表达式：

　　()：括起来的部分就是要提取的。

　　.：匹配任何字符串。

　　+：一次或多次。

　　?：不要太贪婪，在找到第一个匹配项后停止。

c. 模板：用$$引用起来，如果在正则表达式中有多个正则表达式，则可以是$2$$3$等等，表示解析到的第几个值给title。如：$1$表示解析到的第1个值

d. 匹配数字：0代表随机取值，1代表全部取值，通常情况下填0

e.缺省值：如果参数没有取得到值，那默认给一个值让它取。

举例：text/html;(.+?)><meta http-equiv，这个表达式提取出内容：charset=utf-8

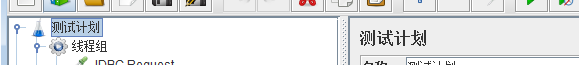
多个内容提取：

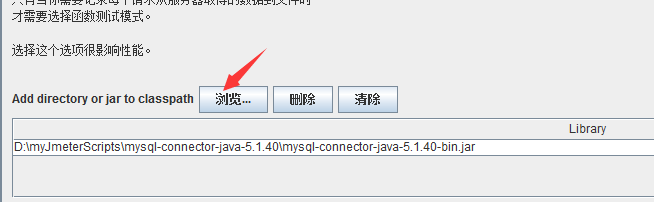
# 10、Jmeter JDBC操作mysql

（1）jdbc.jar包准备，jmeter不能直接连接mysql，需要导入第三方的jar包，来连接mysql；

（2）jdbc jar包导入：

点击测试计划，然后进行jar包的导入





（3）jdbc connect configuration（jdbc连接配置）

选择添加-配置元件-jdbc connection configuration

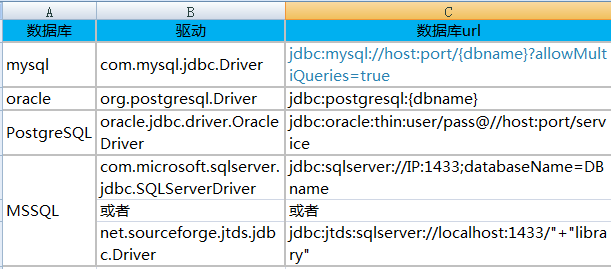
各选项配置：

Variable Name Bound to Pool：必填项，JDBC Request当中引用该变量名，2个地方必须一致。

database URL : jdbc:mysql://192.168.2.25:3306/intellif\_base

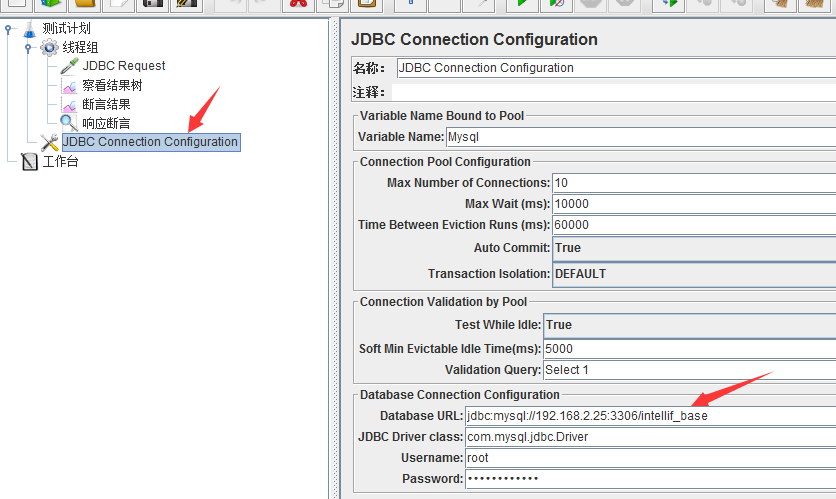
JDBC Driver class : com.mysql.jdbc.Driver

[Jmeter数据库驱动列表](Jmeter数据库驱动列表.xlsx)：

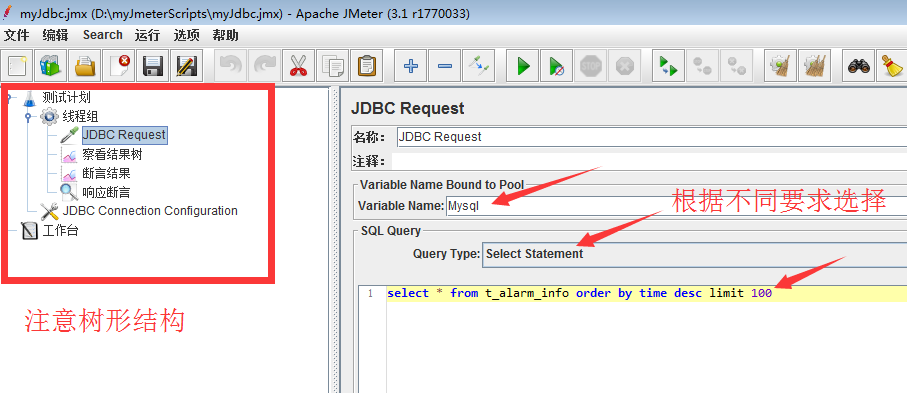


Username：数据库用户名

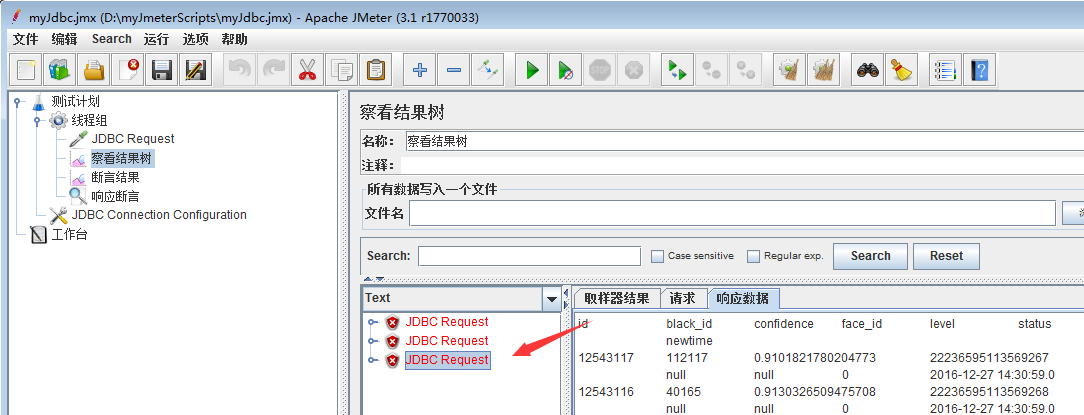
Password:：数据库密码



1. 线程组下添加JDBC Request，注意测试的文件目录结果，如上图左侧所示。



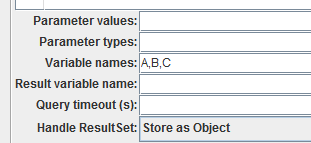
1. 在线程下，添加断言/结果，查看执行结果



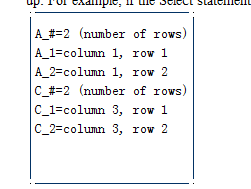
1. jmeter其他请求如何使用jdbc request返回的数据

第一种方法：定义Variable name，然后通过规则来获取数据

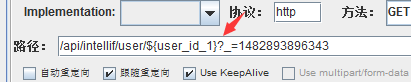
1. 根据要获取的列数值，定义变量个数，例如2个变量：



则取值方式为：

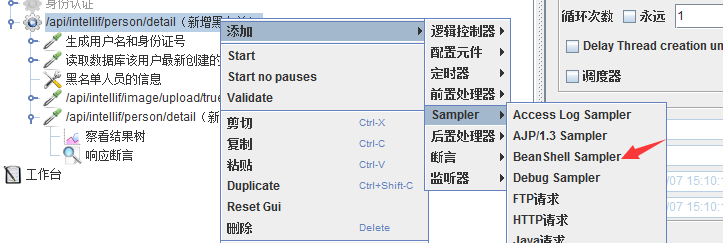


具体调用举例：



注意：jdbc request必须写在http请求之前，线程执行有先后顺序，必须确保调用线程在被调用的线程之后。

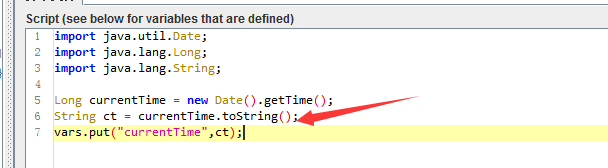
第二种方式：通过BeanShell来编写代码，选择线程-添加-Sampler-BeanShell Sampler获取数据。



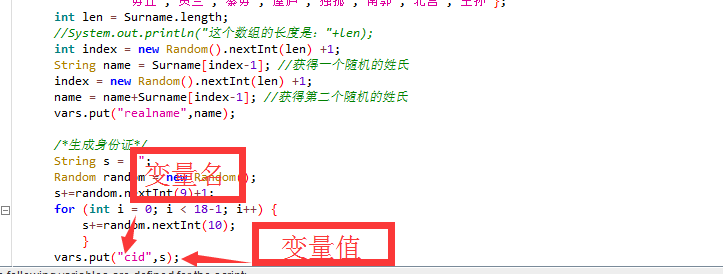
注意：目前发现，beanshell传递的变量，只能是字符串，所以在赋值之前，都需要将变量值转换成string，如下所示：

转换方式：String ct = currentTime.toString();

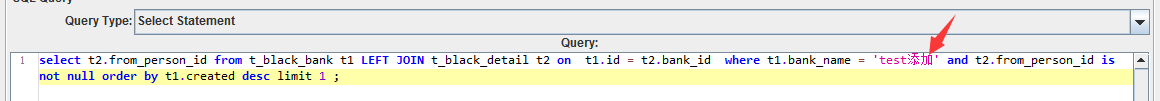
目前发现，在beanshell当中，不支持List操作，会报错。（遇到问题用数组代替）



另外，beanshell遵从java语法，可用java进行实现。



Jmeter查询当中包含中文，查询不返回结果。



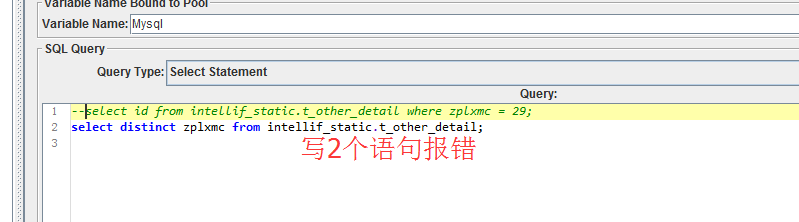
解决方法：

### 2017-02-28目前尚未找到解决的方法，等待后续进行解决。

### Jmeter包含多个数据库

目前的解决方法是，我进行多个jdbc connection configuration

### 为什么jdbc一个request当中，只能写一个select语句。



# 11、压力测试

（1）压力测试的场景测试

单场景：一个请求，接口功能测试

多场景：涉及业务流，有多个请求，多个接口互相调用传参

（2）压力测试数据准备

造数据的2种方法：

Jmeter读取数据库，来获取数据

Python连接数据库，批量造数据：这个目前是不会的。

（3）压力测试结果查看

Tps：服务器每秒处理请求的个数，tps越高，说明服务器性能越好。

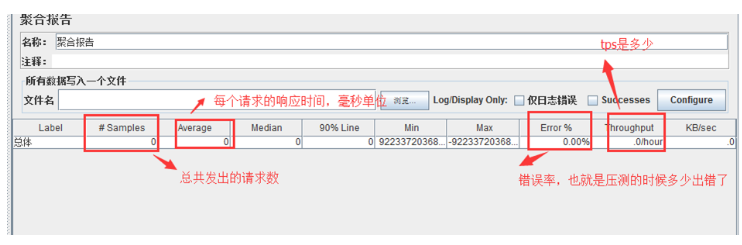
响应时间：每个请求的处理时间。

关发用户数：并发数。

（4）查看方式

通过查看聚合报告，来查看具体的结果，并进行分析。

选择线程组-添加-监听器-聚合报告



# 12、Jmeter添加负载机

应用环境：并发大，一台机器无法满足要求，使用多台机器一起来做并发压测。

添加压力测试机方法：压力测试机上启动Jmeter的代理，然后作为控制机的机器添加压力测试机的IP。

具体操作步骤：

1）、Jmeter的bin目录下有一个jmeter-server，启动它即可，

windows机子作为压力机的话，运行jmeter-sever.bat，linux作为压力机的话，运行jmeter-server

2）、然后在作为控制机的jmeter配置文件里添加压力机ip即可，

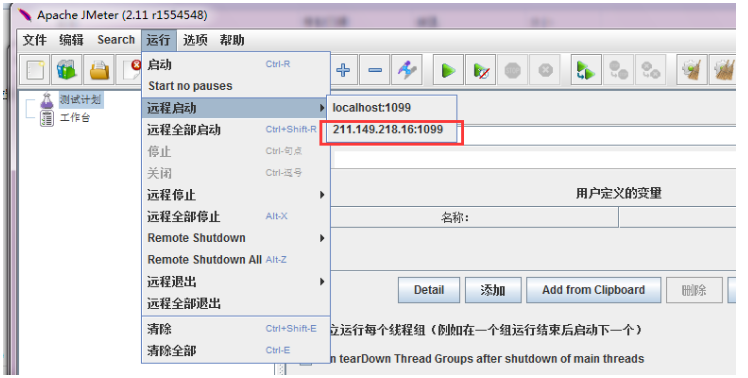
jmeter的配置文件在bin目录下jmeter.properties，

添加压力机的时候

，在配置文件里面找到remote\_hosts添加ip，

然后重启jmeter就可以看到远程压力机了后重启jmeter就可

3）、在配置文件里面添加之后，重启jmeter，在运行的时候，就可以看到压力机的ip了，可以选择他来运行这个脚本，全部运行的话，也就是所有压力机上的全部都启动，就选择远程全部启动，就所有的压力机就都运行了。



# 13、jemter设置识别中文

有2种方式：

（1）进入jmeter的bin目录，找到文件：jmeter.properties，直接修改sampleresult.default.encoding=UTF-8

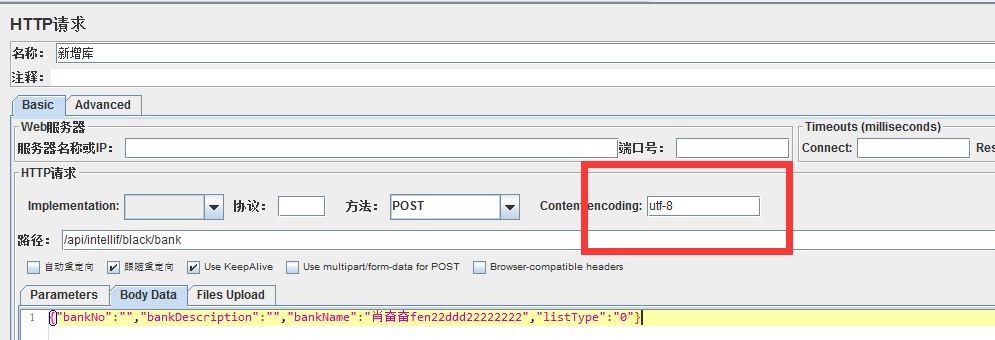
（2）找到要设置的HTTP请求，添加-后置处理器- BeanShell PostProcessor，其脚本框中输入：prev.setDataEncoding("UTF-8");，即可完成设置，这种方法是进行指定布局的设置，（1）中的方法是全局设置。

Jmeter写入数据库当中的中文为乱码

问题原因：接口请求参数值中有中文，http请求中该参数未编码

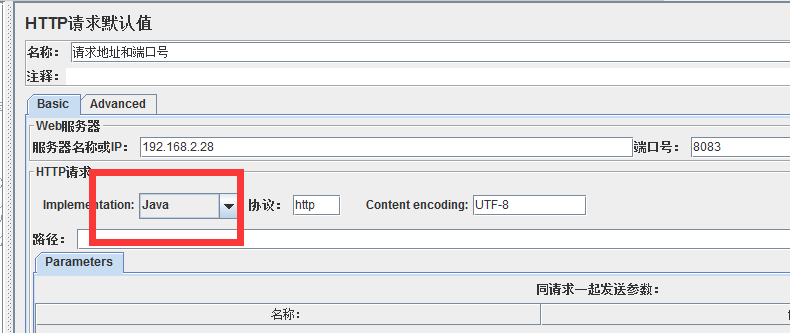
解决方法：一般在请求方指定数据的编码格式，然后再接收方也指定同样的数据编码格式来接受数据。

如下所示：接口请求时，需要进行编码，避免乱码问题出现



问题2：jmeter put方式请求时，请求和响应中，中文显示为乱码。

解决方案：



# 14、几种请求方式的区别

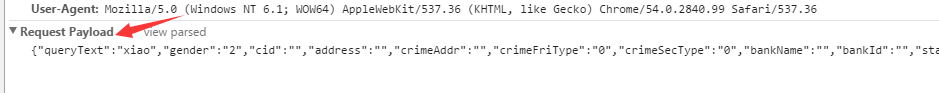
（1）get方法

（2）post方法

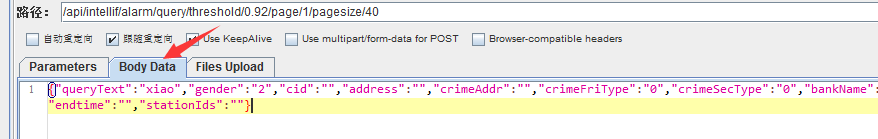
传参的方式不同

第一种：参数，以json格式，在body data里面进行传递。

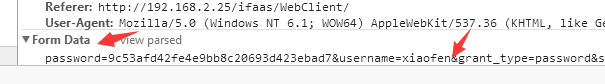
如：



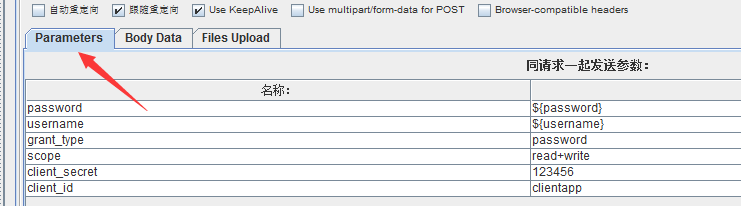
在jmeter当中：



第二种：参数，以&拼接，单独给出每个参数



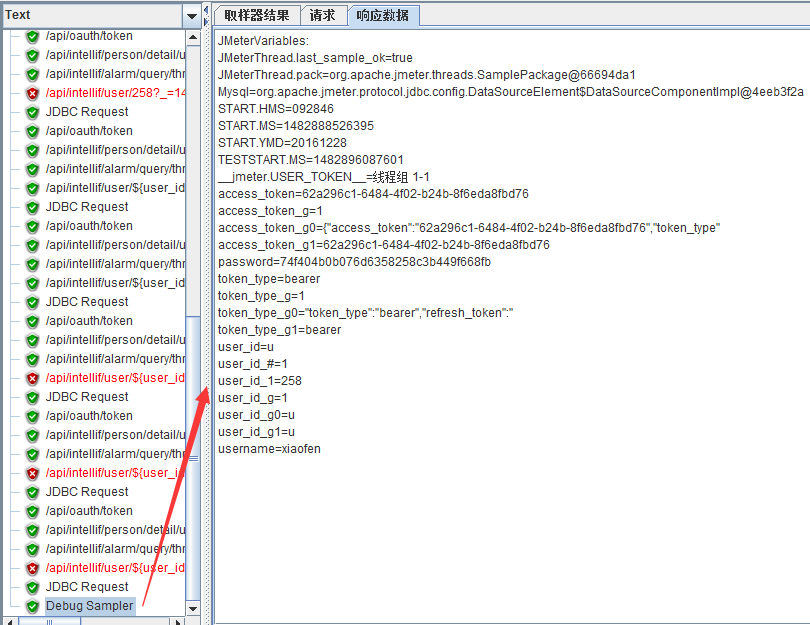
在Jmeter当中：



# 15、debug sampler调试器

（1）在线程组上选择添加-sampler-debug sampler，可以添加1个调试器；

（2）在线程组的结果树当中，可以看到当前运行中的所有的变量参数的取值，如图所示；



# 16、jmeter元件作用域

Jmeter的所有元件，都是有其对应的作用域的，在添加时，需要注意其作用域以及执行的先后顺序。

1. 元件执行顺序规则

a、配置元件（config elements ）

b、前置处理程序（Per-processors）

c、定时器（timers ）

d、取样器（Sampler）

e、后置处理程序（Post-processors） （除非Sampler 得到的返回结果为空）。

f、断言（Assertions）（除非Sampler 得到的返回结果为空）。

g、监听器（Listeners）（除非Sampler 得到的返回结果为空）。

（2）元件作用域

配置元件（config elements ）: 元件会影响其作用范围内的所有元件。

前置处理程序（Per-processors）:元件在其作用范围内的每一个sampler元件之前执行。

定时器（timers ）:元件对其作用范围内的每一个sampler 有效

后置处理程序（Post-processors）:元件在其作用范围内的每一个sampler元件之后执行。

断言（Assertions）:元件对其作用范围内的每一个sampler 元件执行后的结果执行校验。

监听器（Listeners）：元件收集其作用范围的每一个sampler元件的信息并呈现。

# 17、excel用例参数化

# 18、集合点

# 19、jmeter的一些内置函数以及变量的特殊说明

Python中有其自带的一些函数，函数的引用方式如下：

如，调用函数threadNum来获取线程数：${\_\_threadNum()}

还有其他的一些函数：具体请参考以下：



# 20、jmeter输出好看的运行结果

（1） Jenkins 上安装 “Publish performance test result report” ，“Publish HTML reports” 的插件，配置一下。

（2）性能测试报告

（3）Ant,maven工具

# 21、接口持续集成

学习地址：https://testerhome.com/topics/2580

我提供一个build.xml文件，我也是Jenkins+Jmeter+SVN实现接口测试

方案：定时跑接口，输出报告结果（并且邮件通知）

输出性能测试报告

绘制性能图

# 22、jmeter能否设置线程组的调用先后顺序

可以，在测试计划里勾选上：独立运行每个线程组（例如在一个组运行结束后启动下一个），即可要求所有线程组按照我自己添加的顺序，顺序执行



# 23、Beanshell的使用

## 1、beanshell的介绍

A、java编写，符合java语法规范的脚本语言，又拥有自己的一些语法和方法；

B、分类

定时器：　　BeanShell Timer

前置处理器：BeanShell PreProcessor

采样器：　　BeanShell Sampler

后置处理器：BeanShell PostProcessor

断言：　　　BeanShell断言

监听器：　　BeanShell Listener

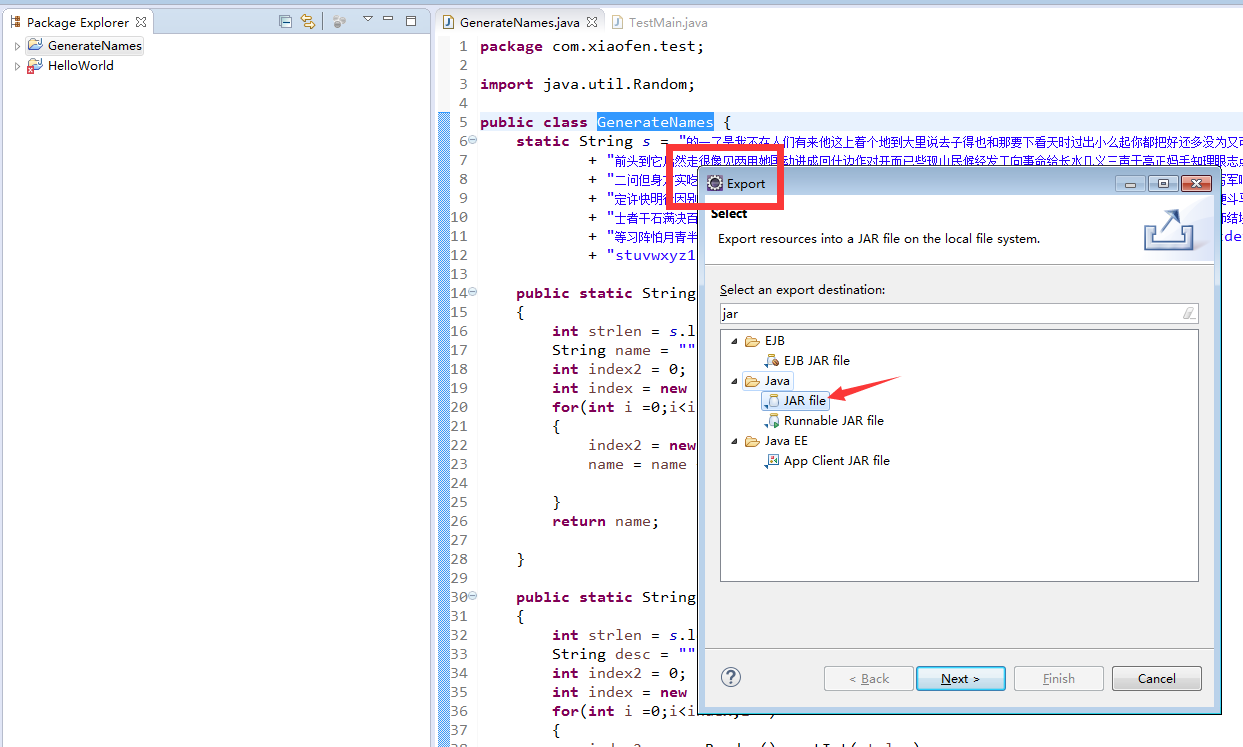
## 2、具体用法

用法一：

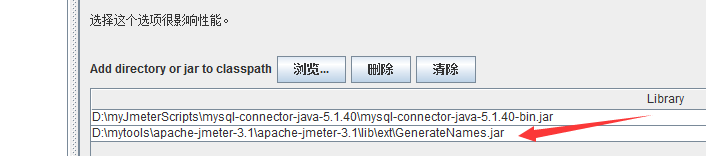
1. 在eclipse编写代码，写好类，导出.jar包；
2. 将导出的.jar包放入到jmeter目录\apache-jmeter-2.13\lib\ext下。
3. 在beanshell当中导入需要的.jar包，调用里面的方法，将结果保存在jmeter变量当中，供后续的使用。

实例操作：

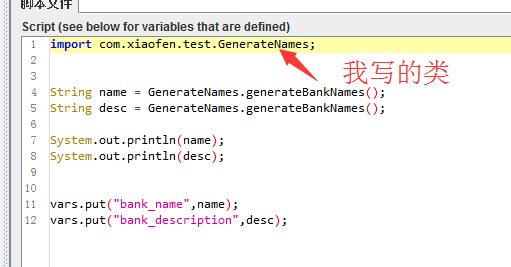
1. eclipse编写生成库名和库描述的类，导出jar包；



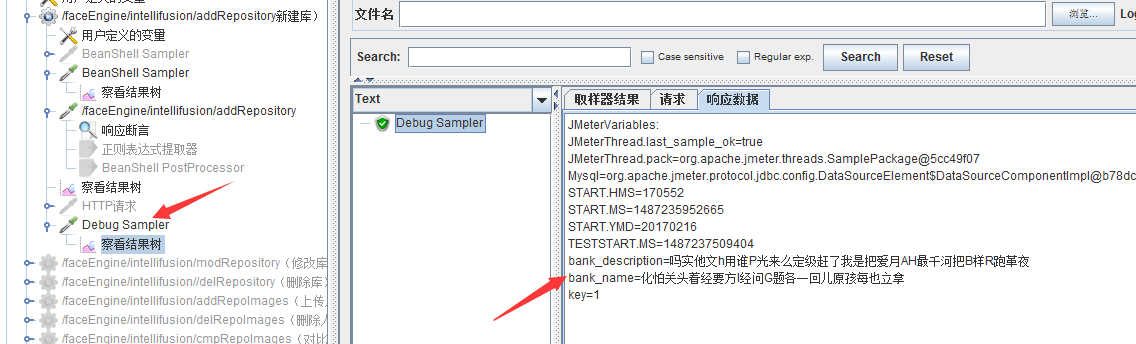
1. 提供给jmeter使用，jmeter利用其生成库名和库描述；（将导出的.jar包放入到对应的目录），并且在jmeter项目中导入jar包。



1. 在jmeter的beanshell当中引用jar包中编写的函数和类；



1. 添加debug Sampler查看运行中各变量的值

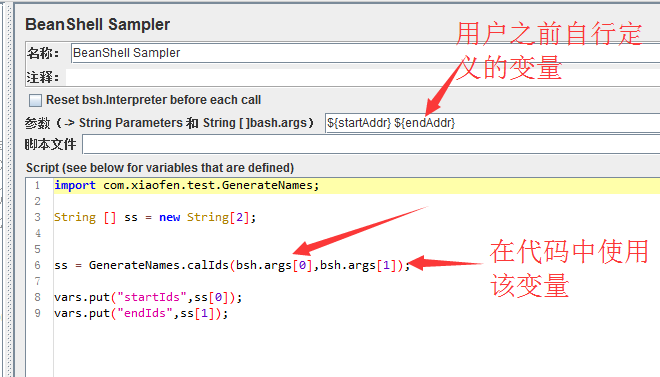


## 3、如何将变量传入到beanshell当中

通常在beanshell当中，需要引用用户自行定义的变量，这样就不需要用户每次来此修改代码，变量的传入和引用方式为：

Beanshell可以接受传入参数：如：${startAddr} ${endAddr}

参数可以通过bsh.args[]按顺序提取



## 4、导入.class

也可以通过导入.class文件的方式，导入某个类，然后调用类中的方法进行测试，如代码：

addClassPath("D:\\javaProjects\\GenerateNames\\bin\\com\\xiaofen\\test");

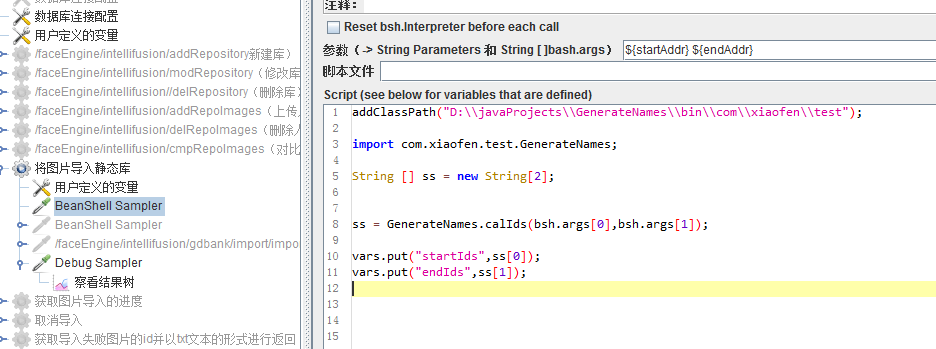
import com.xiaofen.test.GenerateNames;

String [] ss = new String[2];

ss = GenerateNames.calIds(bsh.args[0],bsh.args[1]);

vars.put("startIds",ss[0]);

vars.put("endIds",ss[1]);



## 5、Beanshell的常用的内置变量

1. 写日志：log.info("There is a error,xiaofen");

日志写入目录，bin/jmeter.log当中；

1. 操作变量：

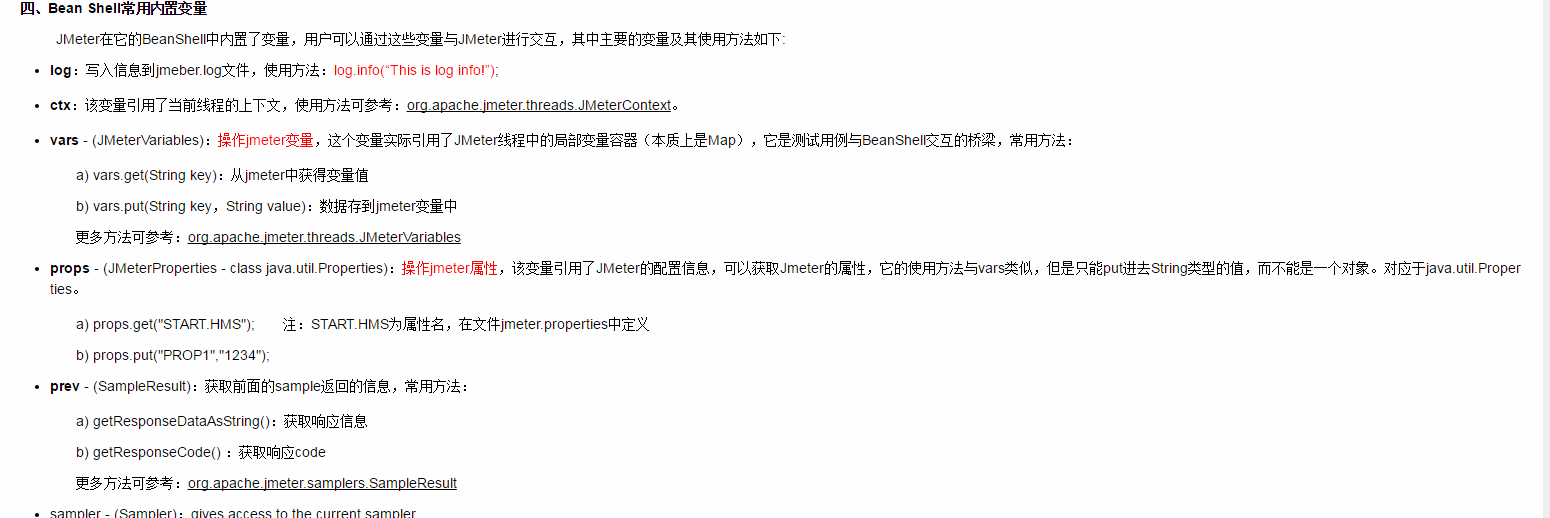
vars.get(String key)：从jmeter中获得变量值

vars.put(String key，String value)：数据存到jmeter变量中

1. **prev** - (SampleResult)：获取前面的sample返回的信息，常用方法：

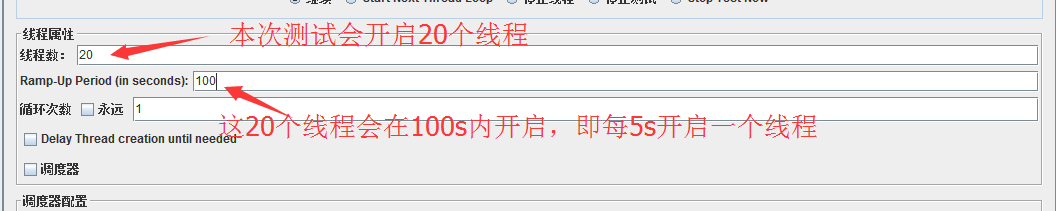
a) getResponseDataAsString()：获取响应信息

b) getResponseCode() ：获取响应code



# 24、 [jmeter通过jmeter-plugins、jconsole对系统资源进行监控](http://blog.csdn.net/gld824125233/article/details/52800280)

# 25、如何设置线程的间隔时间运行的



# 26、jmeter脚本录制器

Jmeter有自己的脚本录制器，它可以对浏览器中所有的操作记录接口请求进行记录，用户可以通过筛选查看这些信息获取所有的接口和请求等信息。

http(s) Test Script Recorder允许Jmeter在你使用普通浏览器浏览web应用时，拦截并录制你的操作。

1. jmeter有一个根证书，会自行生成的

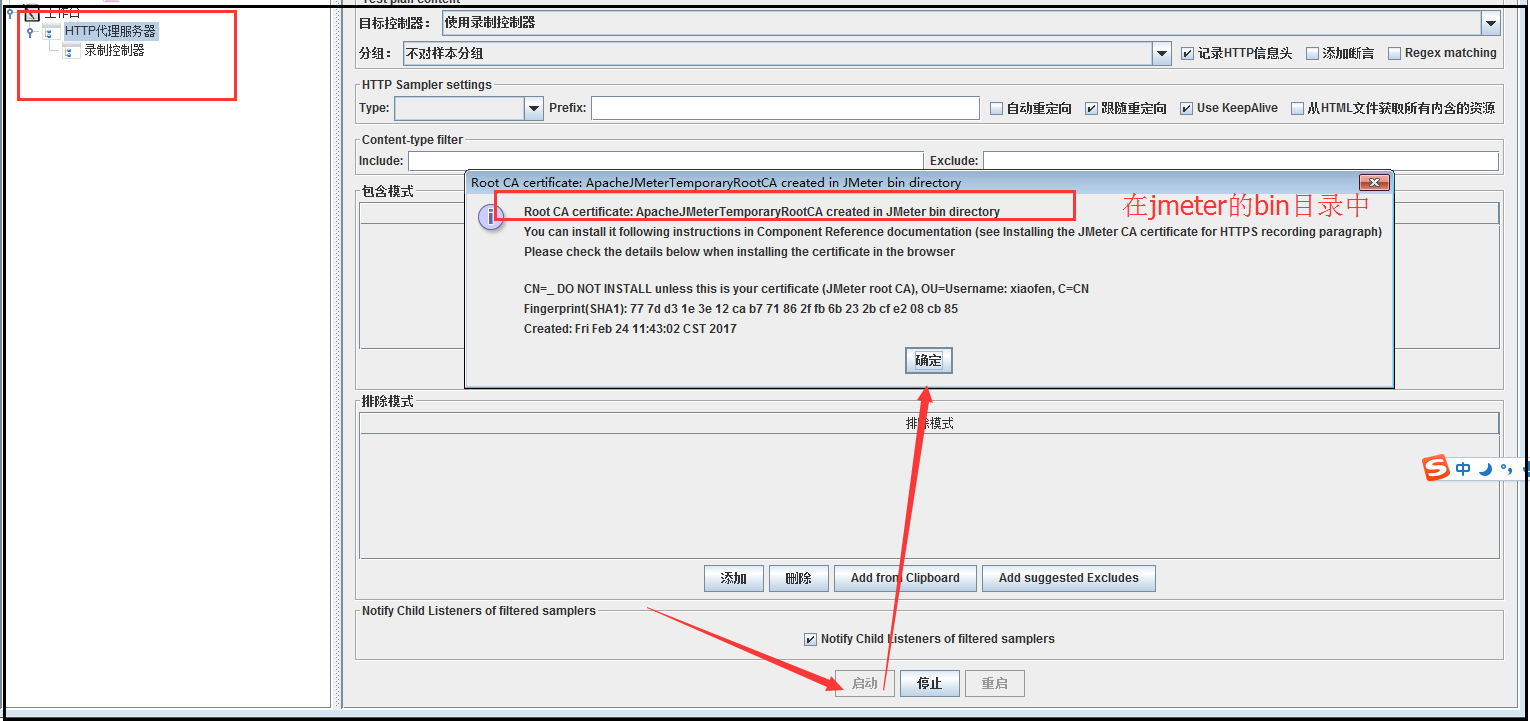
疑问：jmeter的这个证书有什么作用，如果我要测试自己的web网站，这个证书是否与我这个网页有关联，并且我应该如何获取该证书；

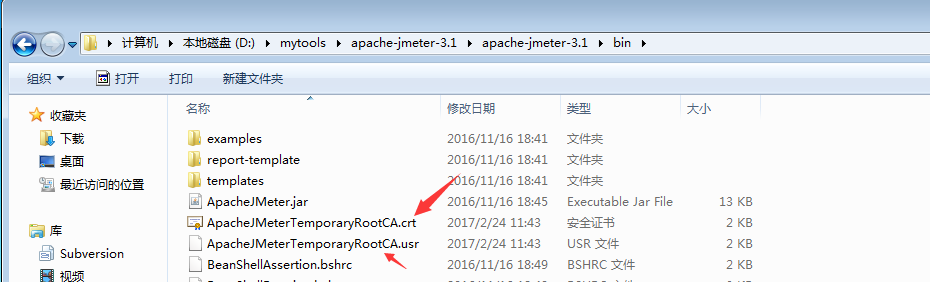
解疑：这个证书添加，应该是让浏览器信任jmeter，之后进行相应的设置，jmeter可以记录浏览器当中，所有的浏览请求记录。

https请求需要客户端与服务器建立签证，如果使用代理服务器，代理服务器没有证书，所以https协议不能正常通信；jmeter提供了相应的证书，可以使https协议正常通信

疑问一：为啥我的的jmeter根目录，没有根证书？

解决方案：在工作台-》添加非测试元件-》添加http代理服务器-》点击启动，即可生成jmeter的证书，如下图所示：

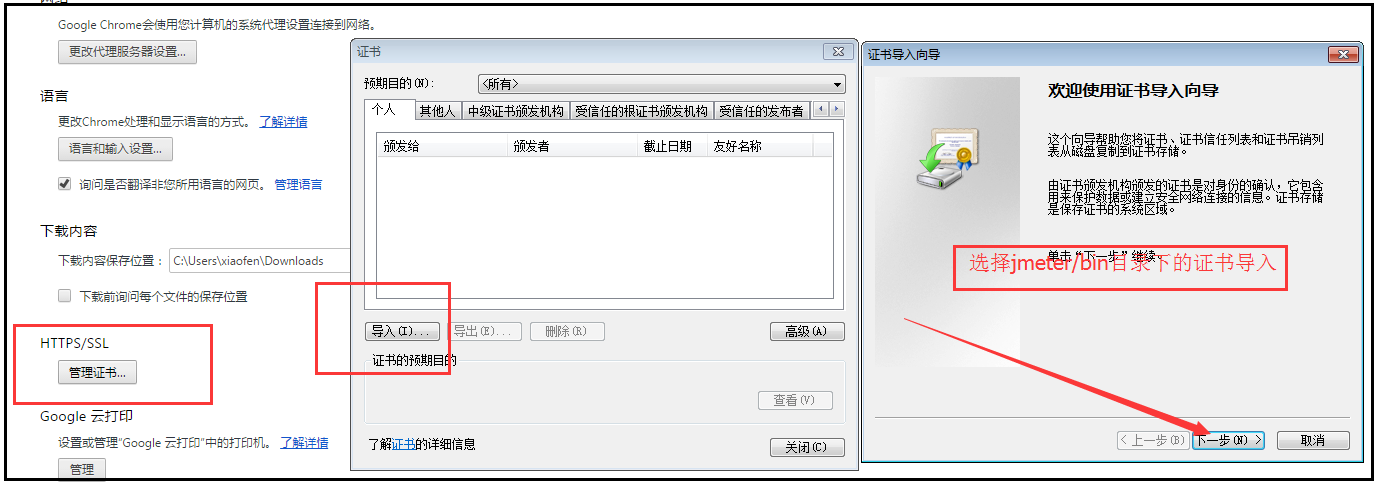




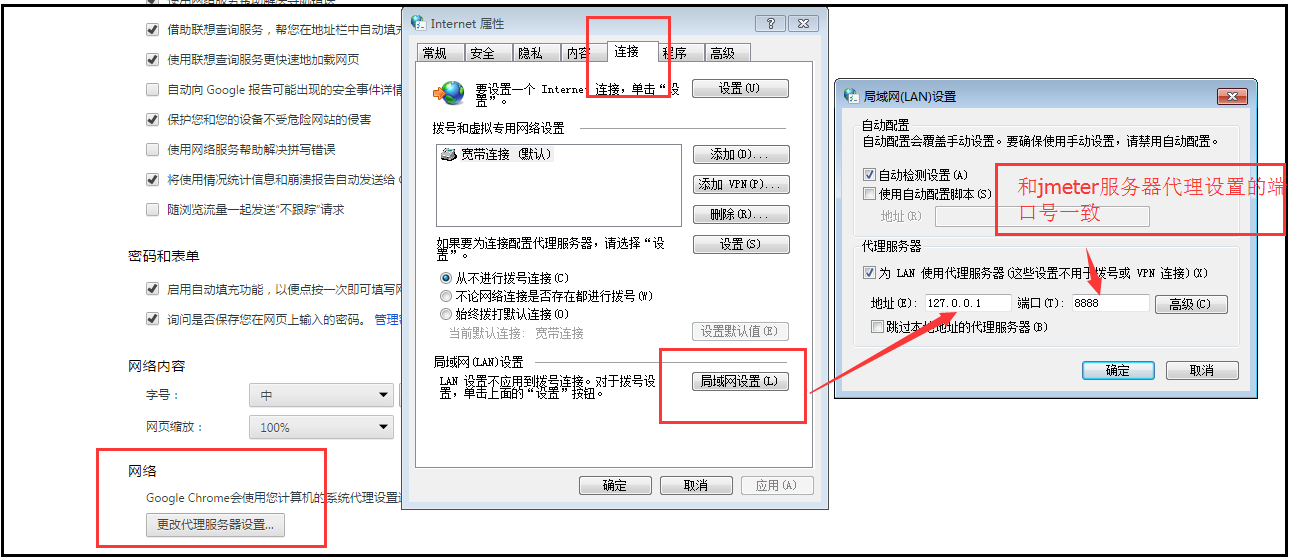
1. 浏览器导入jmeter的证书

导入方式，以chrome浏览器举例说明：

1. jmeter生成证书；
2. 打开chrome浏览器，进入设置-HTTPS/SLL，导入jmeter的证书：

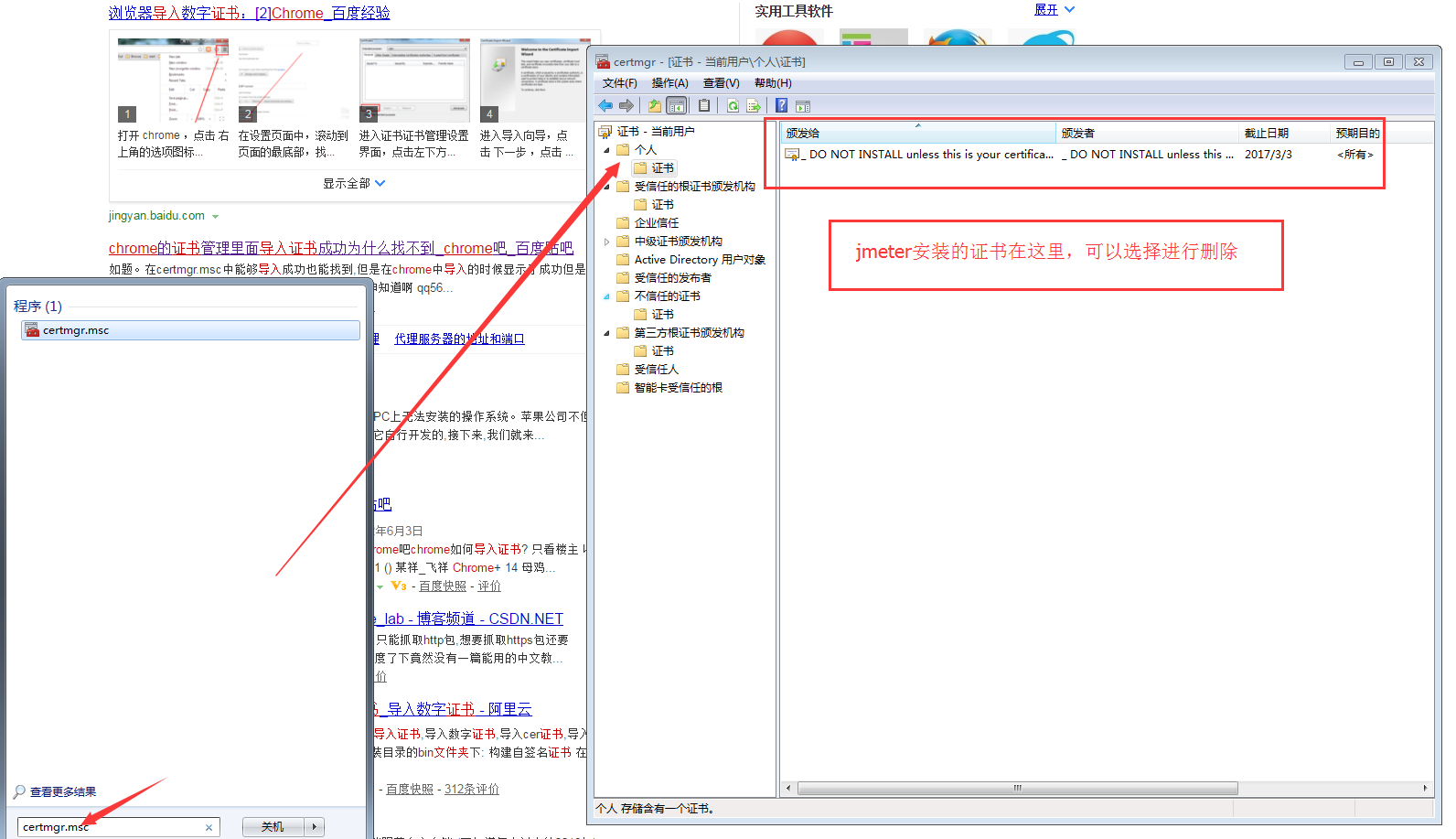


1. 修改浏览器的代理服务器设置：

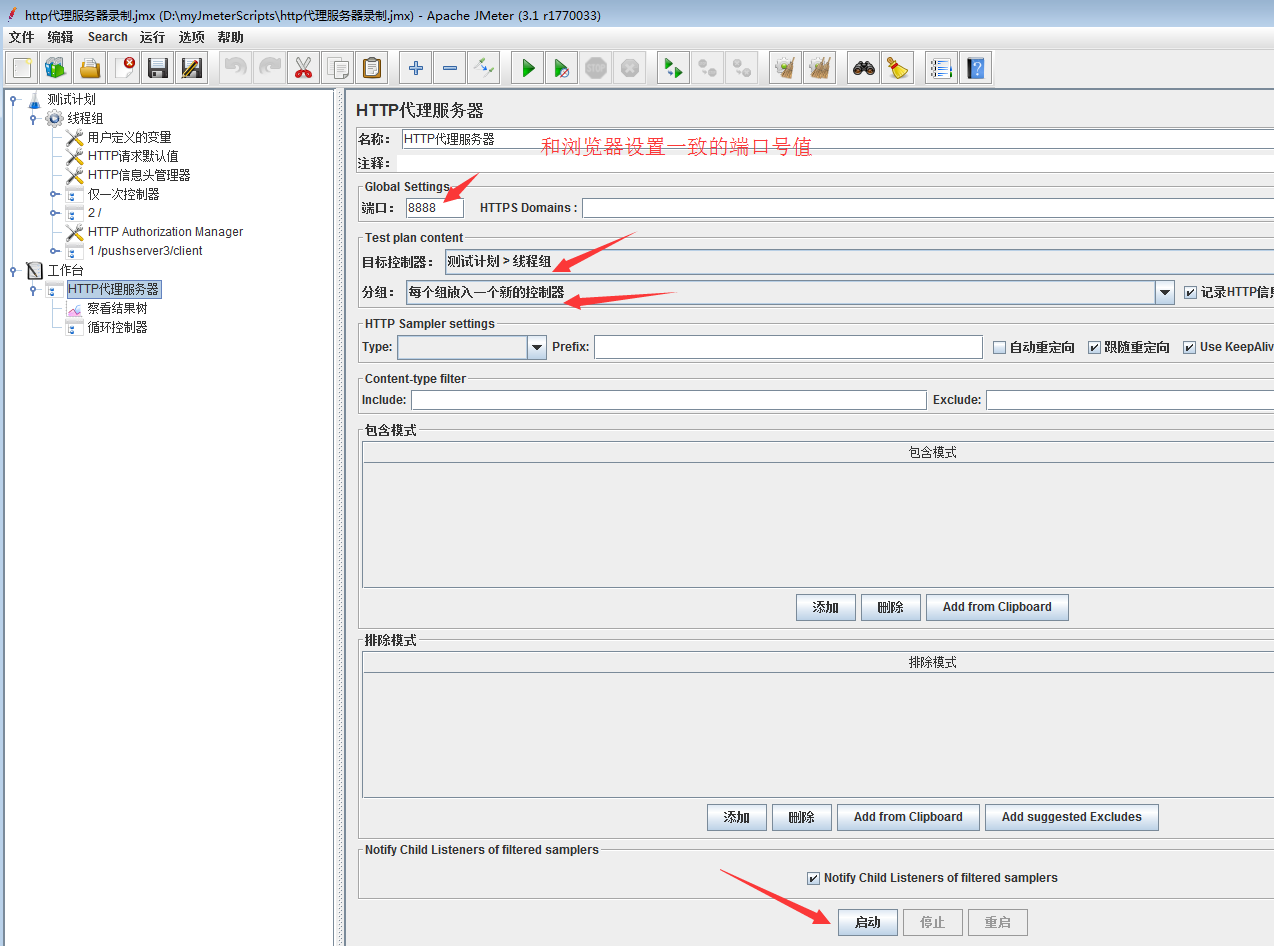


小知识：导入证书后，如果查看已导入的证书，并且将其删除。（certmgr.msc）

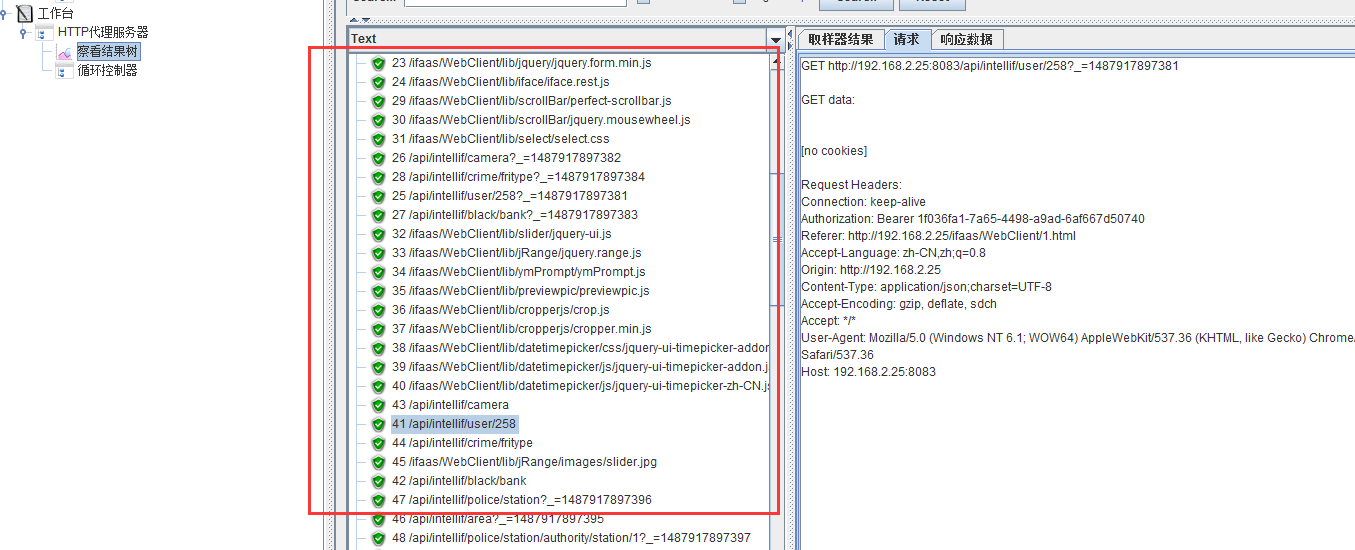
备注：设置浏览器代理后，浏览器会无法进行网络访问，要打开要测试的网站，需要在jmeter当中，启动代理服务器，然后即可进行网络访问了。



（3）启动jmeter的代理服务器，进行录制，在结果树中，可以看到所有的请求记录。



请求的信息：



（4）对录制的结果进行筛选

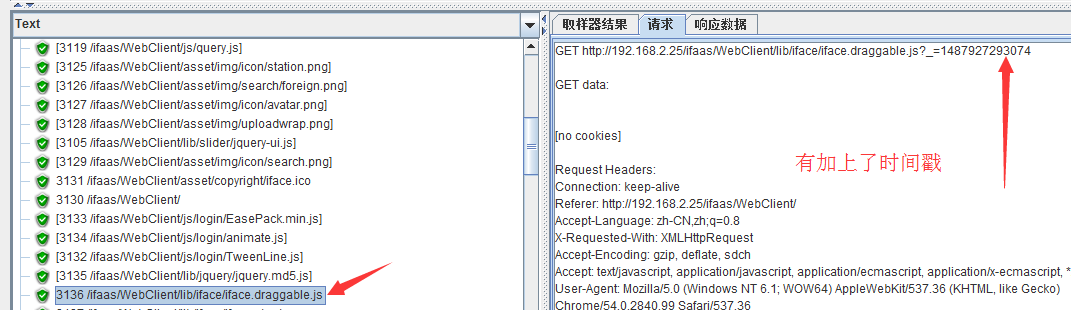
Jmeter的数据筛选，是采用了正则匹配的规范，所以要根据其规则来进行匹配过滤。

1. 过滤掉.js信息

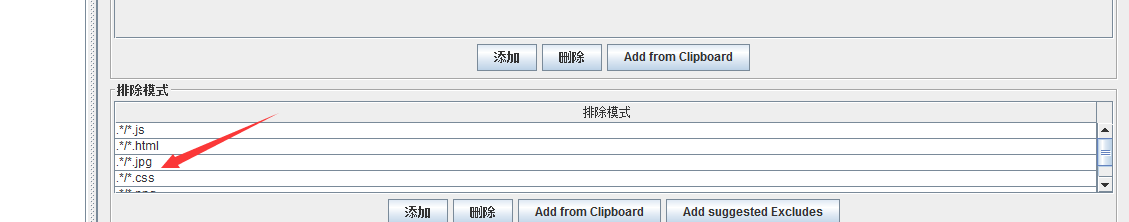
.js好像与其它的过滤方式有所不同，一直无法过滤掉.js。

问题原因：.js确实是被过滤掉了一些，但是对于一些有\_时间戳的请求，没有进行过滤掉。

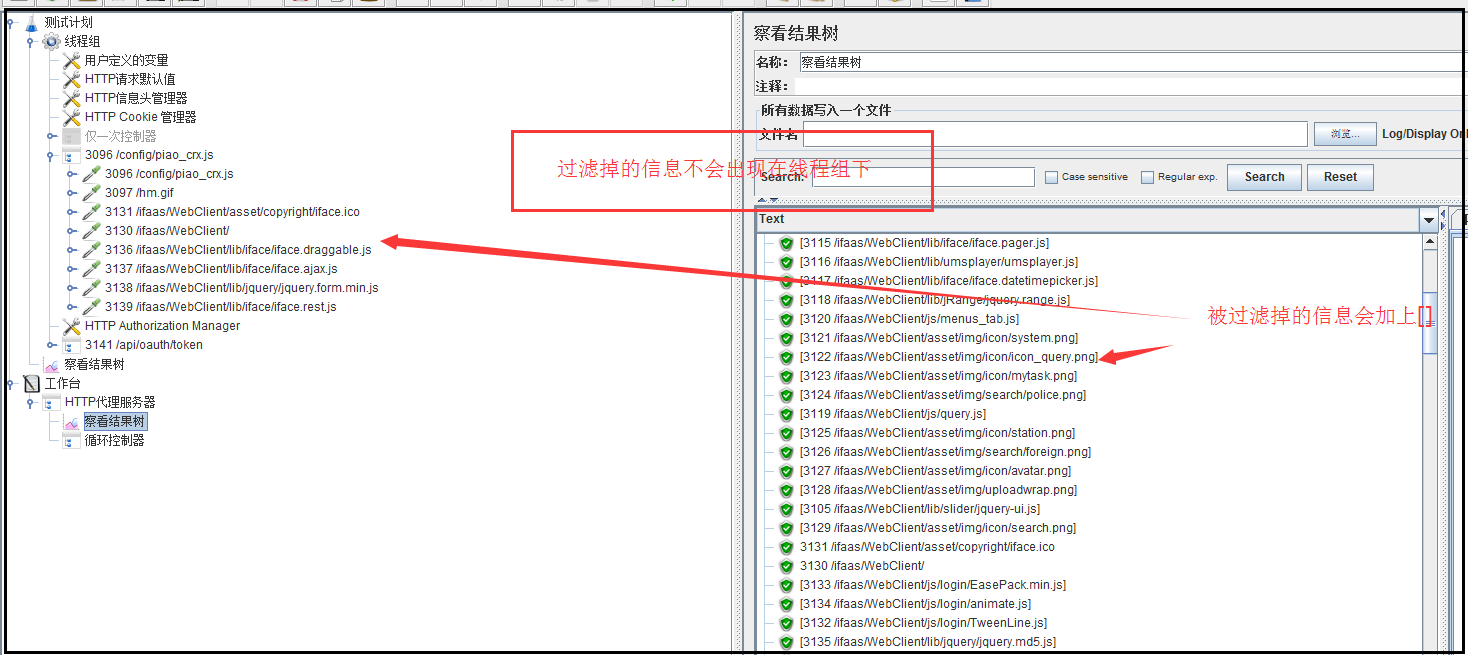
问题解决方案：



1. 其它信息的过滤遵循以下条件：.\*/\*.type



1. 过滤的时候，代理服务器的录制结果树当中，已经过滤掉的信息会加中括号[]，被过滤掉的信息不会显示在线程组下。如下图所示：



# 27、https请求

https请求因为涉及到证书，所以在调用https请求时，与http请求有所不一致，需要进行以下的相关设置，才能保证接口的成功调通。步骤如下：

1. 正常的添加线程组-http头信息和https请求；
2. 设置jmeter代理，设置及证书的生成，请查看：26、脚本录制器 （1）Jmeter有1个证书；
3. 开启jmeter代理

开启方式为将证书，导入到jmeter当中，具体操作步骤如下：选项-》SSL管理器-》选择生成的证书导入即可；

1. 现在已经设置好代理，重新修改下已经创建好的HTTP请求。再最下方Proxy server处添加代理服务器：localhost（本机上搭建的Jmeter代理），端口号就是8080。保存一下整个计划就可以。（这里还具体不知道是什么意思，后续有待理解）

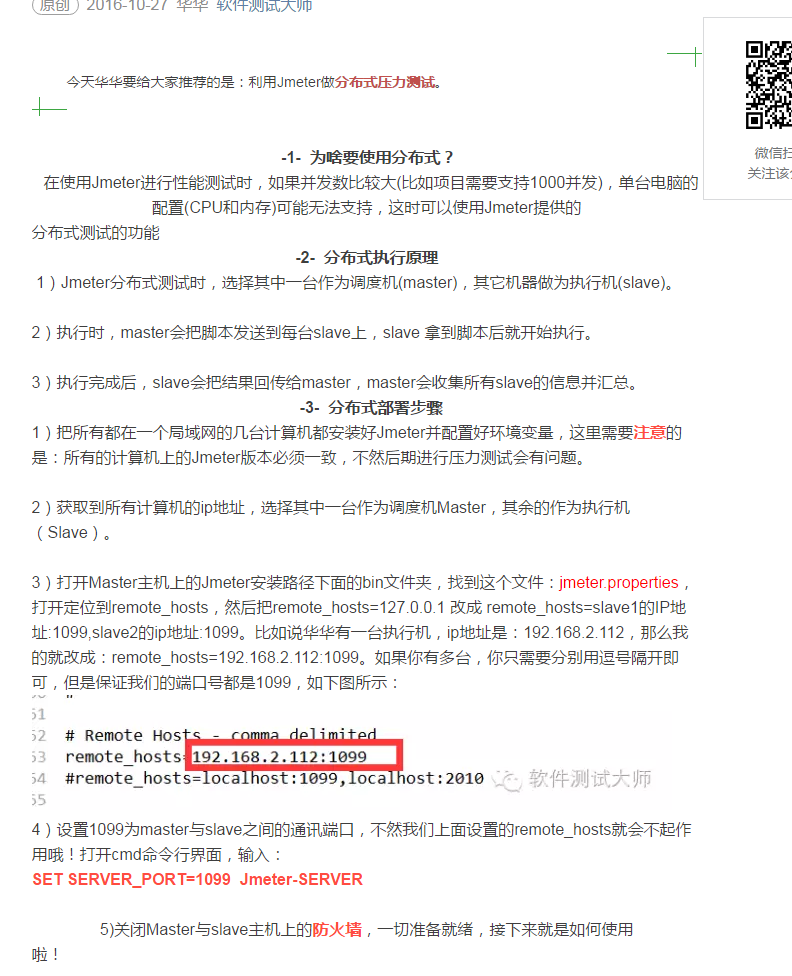


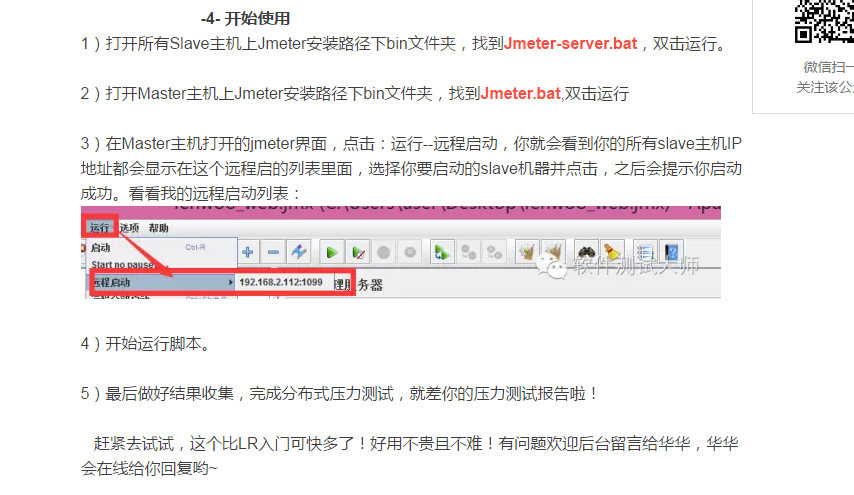
1. 调用https接口查看结果，此时即可调通接口；

在代理服务器处添加查看结果树，运行接口，查看结果。

# 28、Jmeter做分布式压力测试

<https://mp.weixin.qq.com/s?__biz=MzAwNjY4MjE3NQ==&mid=2650427363&idx=1&sn=61912f860e6b4d7f6bf0ba67a0f8abba&chksm=83074cb3b470c5a505b54b6520880646f9063ef804a93a7ab427c41d4c14f9cb70d766dc9688&mpshare=1&scene=1&srcid=0227xfyrUmeU1bA5hN1eKacO&pass_ticket=UPThw%2FMgexgKNBR9CHjHdTAJPg7HIvn9aooR1CkWngVDTT233cKBtla9ktJ9fMHJ#rd>

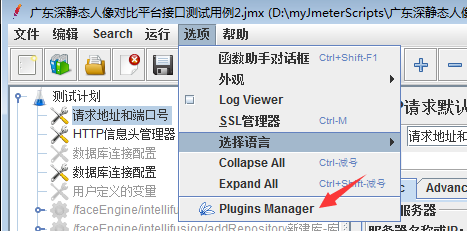


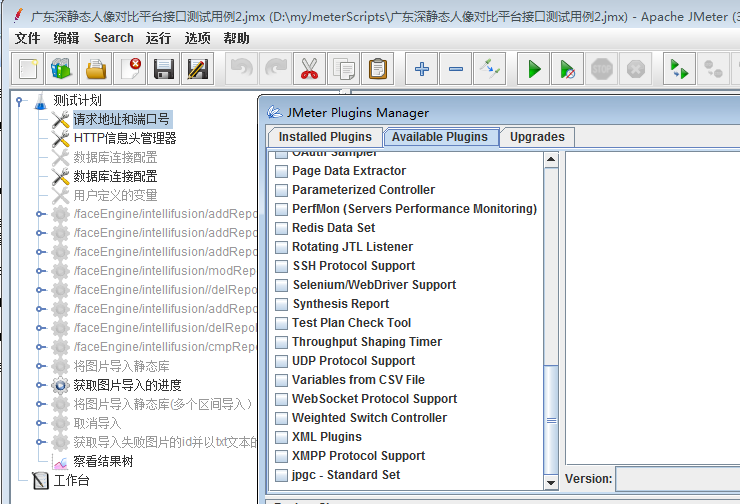


# 29、Jmeter的其他插件

Jmeter当中，有许多可以借助的插件，插件可以通过Plugins Manager来进行管理，plugins Manger的安装步骤如下：

1. 进入jmeter官网，下载Plugins Manager的jar包（下载地址：<https://jmeter-plugins.org/wiki/PluginsManager/>）
2. 将下载的jar包，放入jmeter的安装目录：/lib/ext/目录下，重启jmeter，即可看到安装成功的该管理插件：如下图所示：



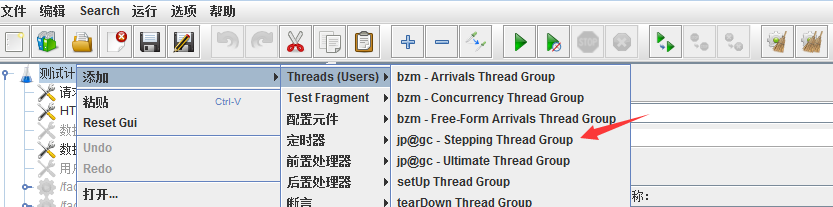


1. 常用的几个插件

插件的安装类似于上面的plugins manager的安装方式，对于plugins manager不能找到的插件，可以通过jmeter官网找到该插件，然后将其jar包放入/lib/ext/目录下即可（注意，每个下载的zip包里面，都包含有一个jmeter-plugins-manager-0.10.jar包，和这个jar包是不需要的，就是plugins manager的jar包）

1. 用于服务器性能监视的PerfMon Metrics Collector
2. 用于建立压力变化模型的Stepping Thread Group

安装成功如下：



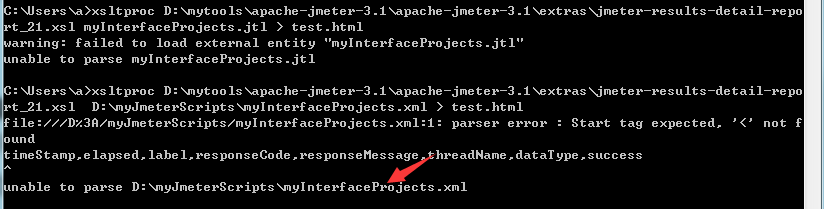
该插件的作用：提高了一些高级的配置选项，用户可以自己建立压力模型。

各参数的意义：

1. This group will start Max threads - 设置单台负载机，线程组启动的线程总数为Max个
2. First,wait for N seconds - 启动第一个线程之前，需要等待N秒
3. Then start N threads - 设置最开始时启动N个线程
4. Next add N1 threads every N2 seconds, using ramp-up N3 seconds - 然后，每隔N2秒，在N3秒内启动N1个线程
5. Then hold load for N seconds - 单台负载机启动的线程总数达到Max之后，持续运行N秒
6. Finally,stop N1 threads ervery N2 seconds - 最后，每隔N2秒，停止N1个线程数。
7. 用于Json解析的JSON Path Extractor
8. 用于展示响应时间曲线的Response Times Over Time
9. 用于展示TPS曲线的Transactions per Second

# 30、 jmeter3直接生成html格式的report

（1）利用xsl模板，利用命令xsltproc 将运行结果保存的.jtl格式的结果文件准转换成html文件。--但目前我是失败的，无法解析该文件。



Xsltproc，是个转换的小工具。

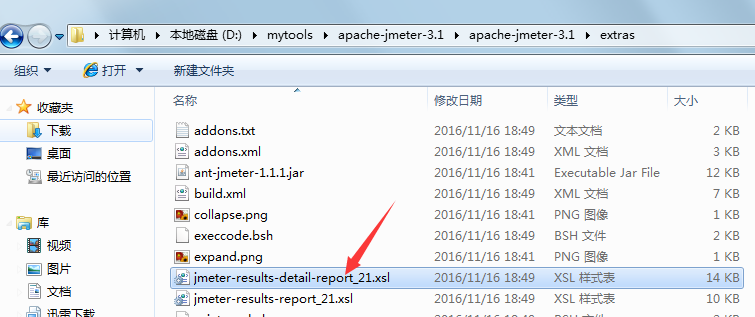


# 31、测试报告模板优化

（1）jmeter自带的测试报告模板（jmeter3有2个）

所在路径：D:\mytools\apache-jmeter-3.1\apache-jmeter-3.1\extras

如下所示：



模板使用：

修改${jmeterhome}/bin/jmeter.profile配置文件，添加所需的报告参数

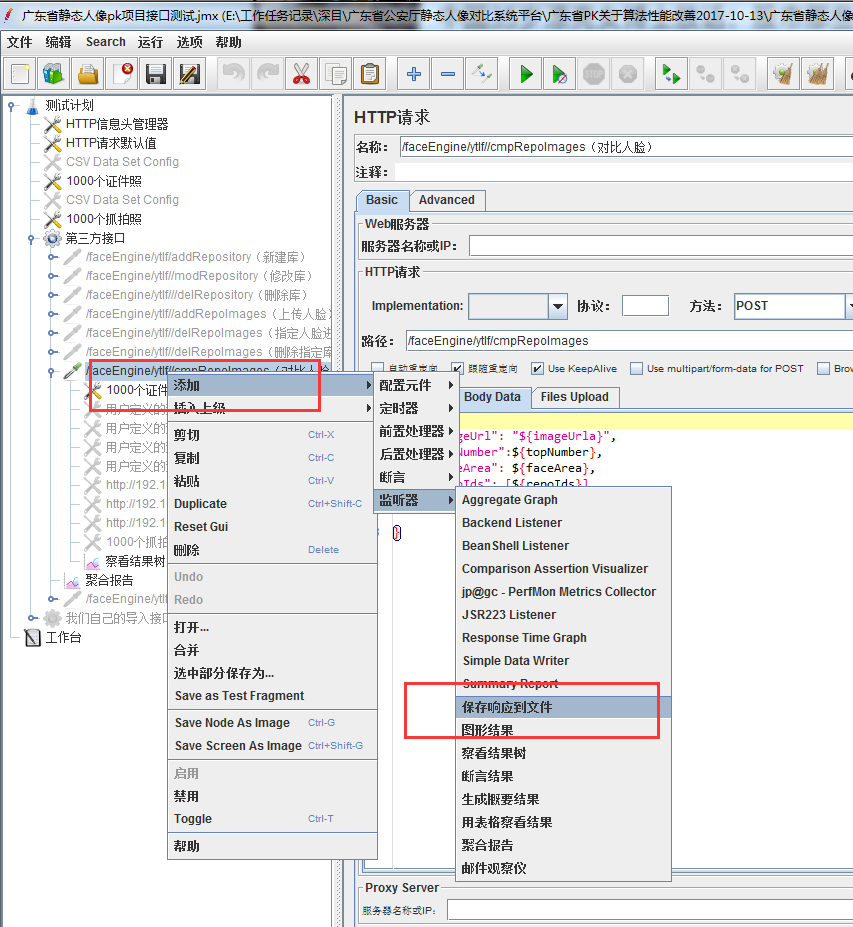
build.xml文件中指定，手动操作学习下xslt工具的使用说明

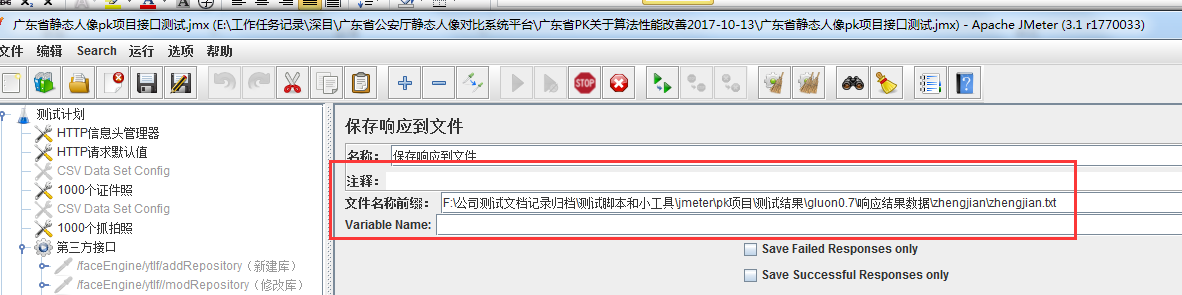
（2）寻找第三方的模板

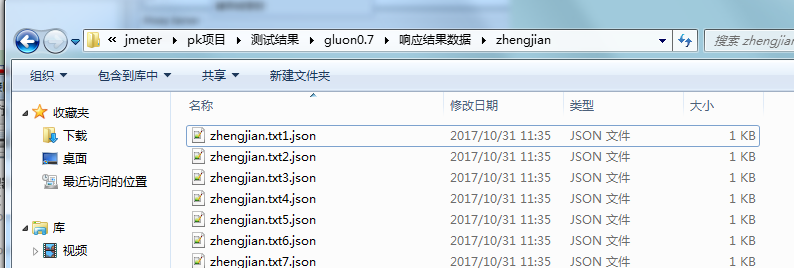
# 32、版本管理之git

（1）安装git插件，用于版本管理

# 33、保存响应数据到文件



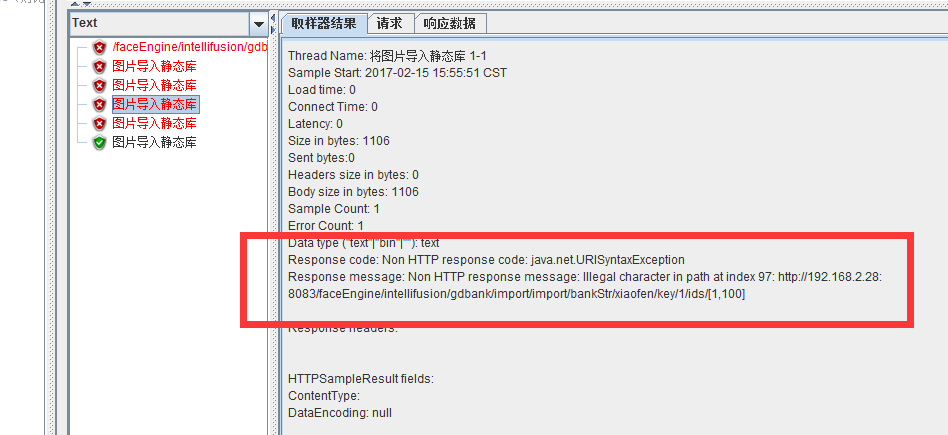




# Jmeter常见问题及解决方案：

## Response code: Non HTTP response code: java.net.URISyntaxException

问题所在：



解决方案：（get/put方式，都应该选择这个java选项）

