操作说明以及问题解决方案，以下是一个事例，请参考。

**操作说明：**学员登录“点考系统”，进入“考试系统”获取试卷进行考试，提交考试，退出“点考”登录。

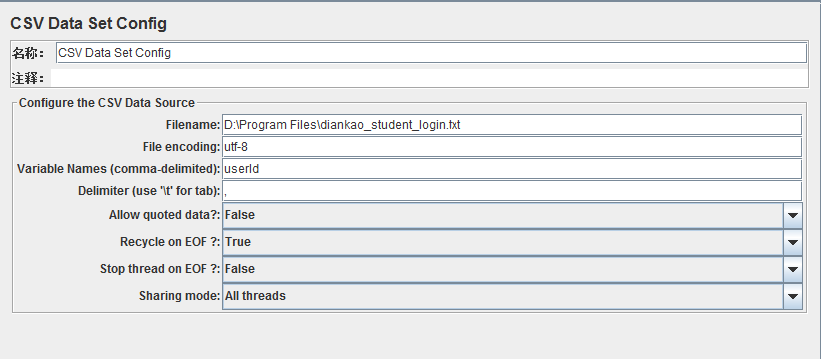
**步骤1：**首先使用badboy录制将要执行的脚本，录制完成，点击File---Export to Jmeter。

**步骤2：**准备学员登录的账号，导入到点考系统中，即学员注册成功；

**步骤3：** 把学员账号与密码导入到一个txt文件中。（此次使用相同的密码，所以只导入了学员账号）；

**步骤4：**使用Jmeter打开脚本，选中“Thread Group”右键---添加---配置元件---CSV Data Set Config。

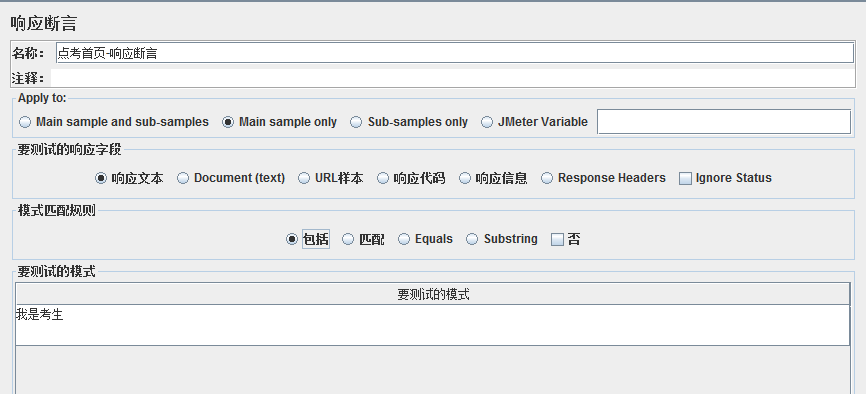
添加内容如下：



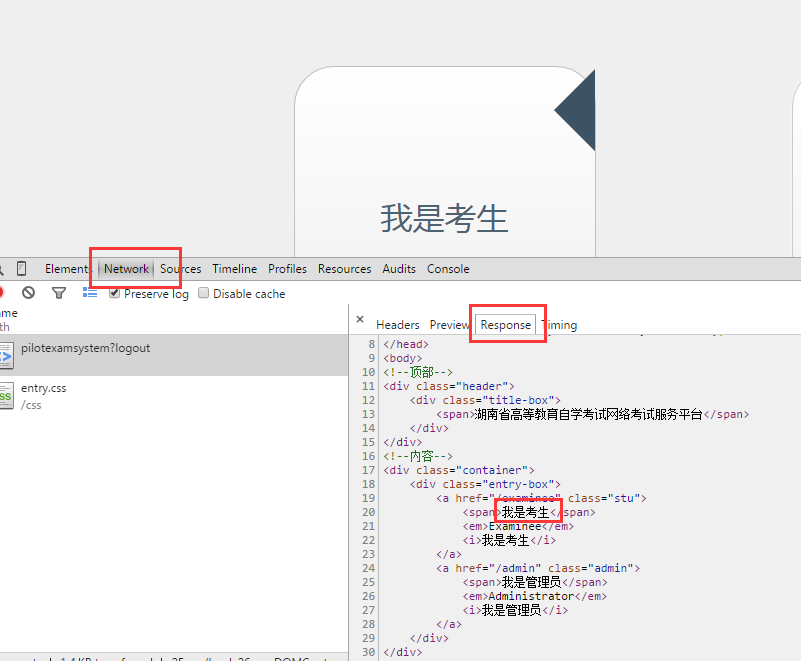
**步骤5：**优化操作步骤，把每一步的名称进行修改。

添加断言，选中需要添加断言的步骤-右键点击“添加”-断言-响应断言。

建议：每一步都需要添加相应的断言，以确认显示的内容是否符合预期结果。



添加“要测试的模式”输入， 根据浏览器F12调出“开发人员工具”-Network-Response中摘取此页面特有字段或语句。



**步骤6：**点击“Thread Group”，当进行优化脚本时“线程属性”均设置为1。



步骤8：选中“Thread Group”右键---添加---监听器---聚合报告。（做性能测试时，

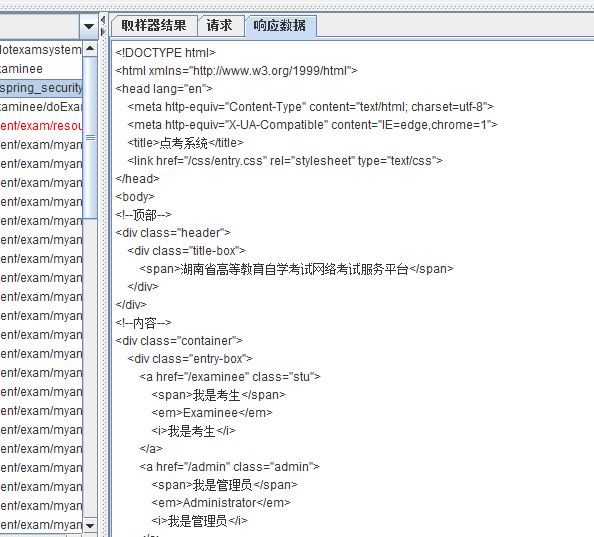
查看速度，后面详情介绍）。

选中“Thread Group”右键---添加---监听器---查看结果树。（参看脚本错误

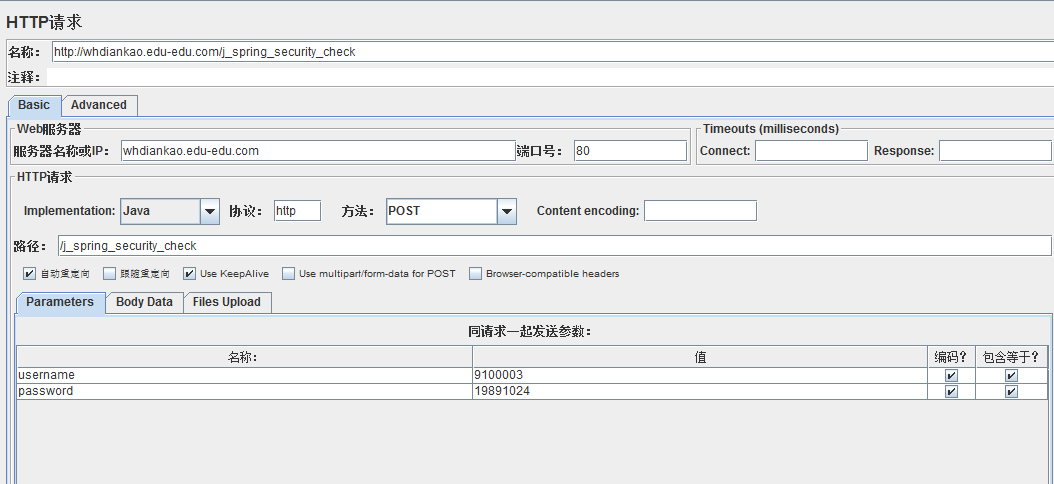
内容）

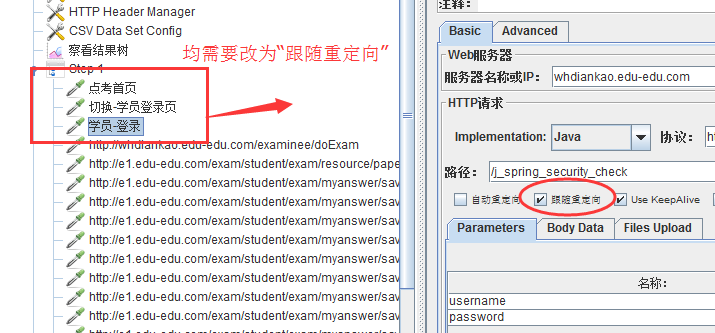
步骤9：根据“查看结果树”执行结果，查看“响应数据”内容与真实的界面显示是否一致。

问题1：此处显然并没有进入到登录后的页面，所以此处有问题。

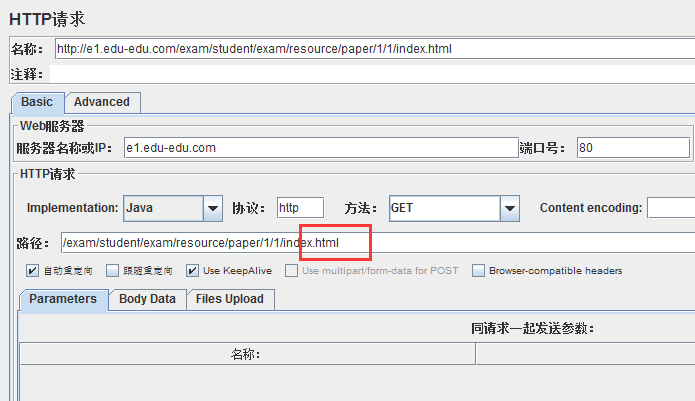


打开相关的HTTP请求页，把“自动重定向”改为“跟随重定向”。包括登录之前的所有步骤均需要改为“跟随重定向”。根据我的理解“自动重定向”是在同一页面做不同的操作，但页面地址不改变。“跟随重定向”是进行了跳转页面。





问题2：当发现有.html的步骤时，说明此处有错误。执行脚本，查看“查看结果树”检查脚本中的错误。

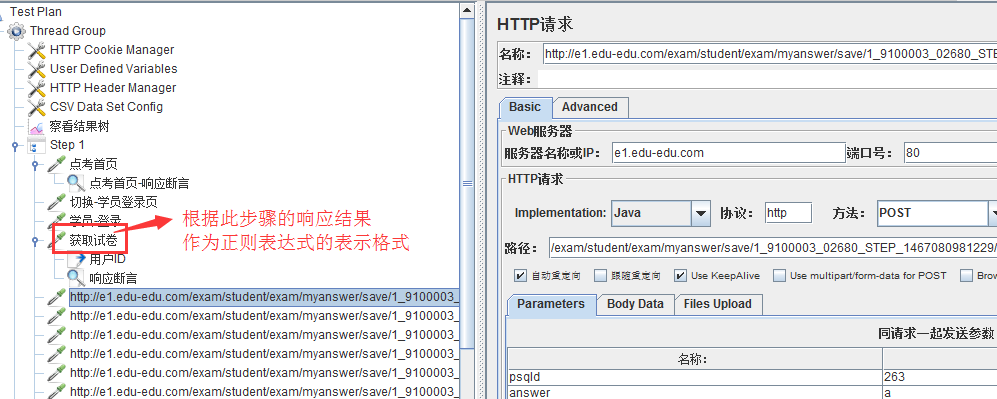


参考此请求的上一步与下一步，如果上一步的请求地址与下一步的请求地址均在一个服务器上，并且运行正确没有出现问题。那么删除此请求即可。（注意：为了避免错误，执行完成后，可以上真实的页面上查看是否有执行的记录，记录内容是否正确）

问题3：由于跳转地址中有些随机改变的数据内容，比如地址：http://e1.edu-edu.com/exam/student/exam/myanswer/save/1\_9100003\_02680\_STEP\_1467080981229/29中“9100003”是会改变的用户账号。所以需要做一个正则表达式来代表这些随机改变的数据，并给它起个名字（比如userId）然后用${userId}引用到下一个页面中。

正则表达式的写法：

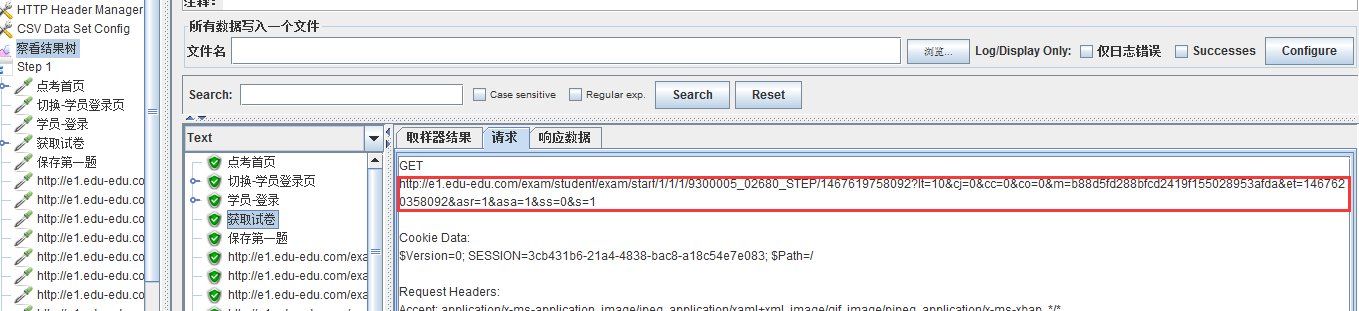
写法的格式规则是根据此HTTP请求的上一步的响应结果内容编写的。



根据运行结果，可知相应结果为

“http://e1.edu-edu.com

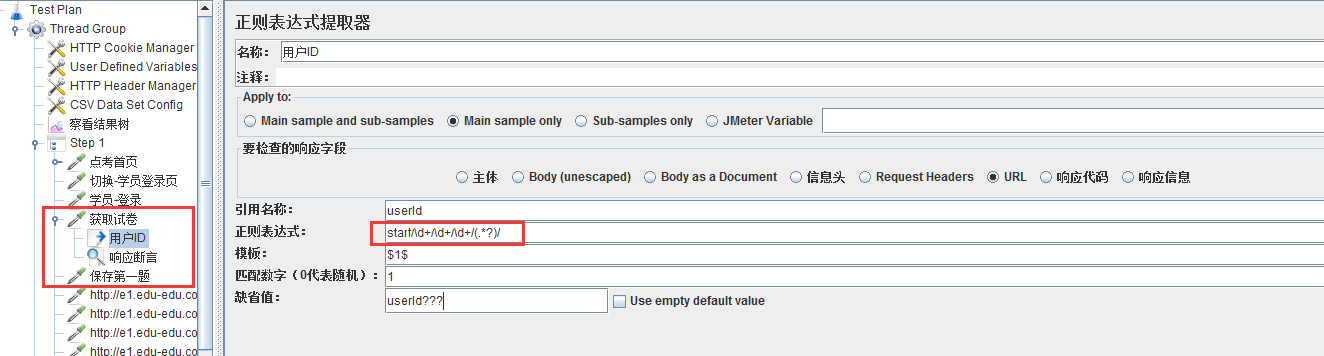
/exam/student/exam/start/1/1/1/9300005\_02680\_STEP/1467619758092?lt=10&cj=0&cc=0&co=0&m=b88d5fd288bfcd2419f155028953afda&et=1467620358092&asr=1&asa=1&ss=0&s=1”



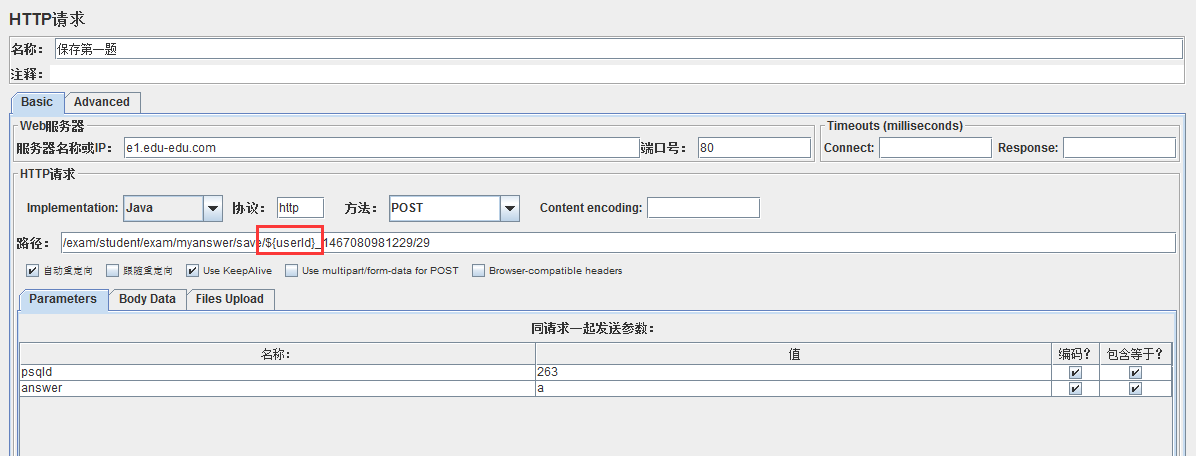
根据这个链接，那么正则表达式为: start/\d+/\d+/\d+/(.\*?)/

* Start为上一步的真实路径
* /\d为路径中的数字
* (.\*?)为需要正则表示的内容。

添加到“正则表达式提取器中”的显示



添加完成后，下一页面中的HTTP请求即可引用这个正则，显示如下：（运行以下，查看引用的是否正确）

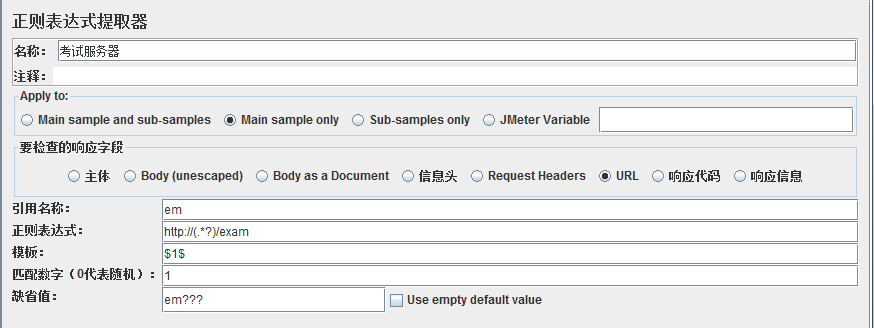


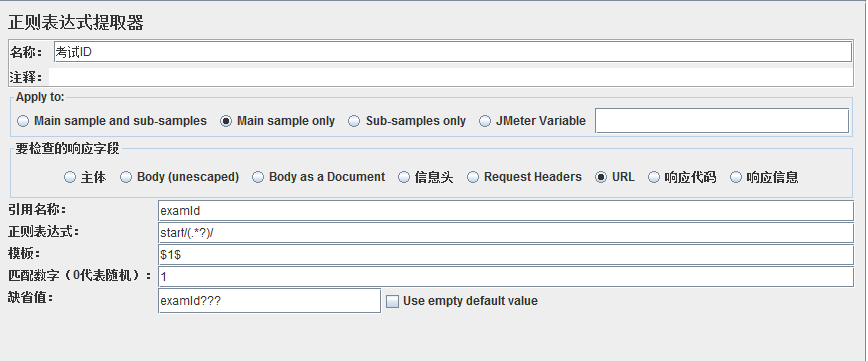
根据以上方法可分别把以下请求中的各项数据分别用正则表示。注意：最好你已经了解这个请求中各项的含义，这样方便定义，也方便后期使用。

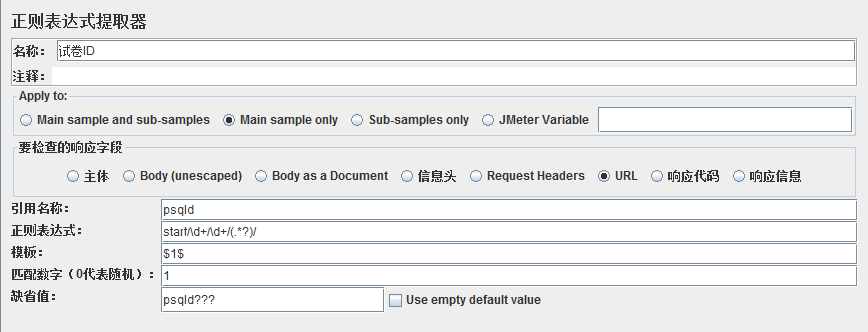
http://e1.edu-edu.com

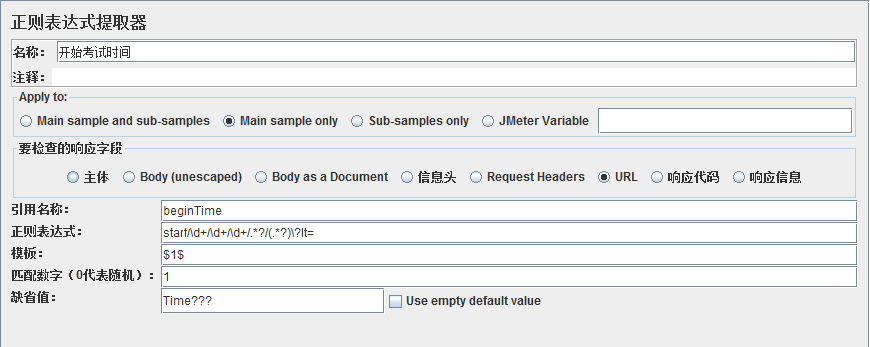
/exam/student/exam/myanswer/save/1\_9100003\_02680\_STEP\_1467080981229/29

相关正则的显示：





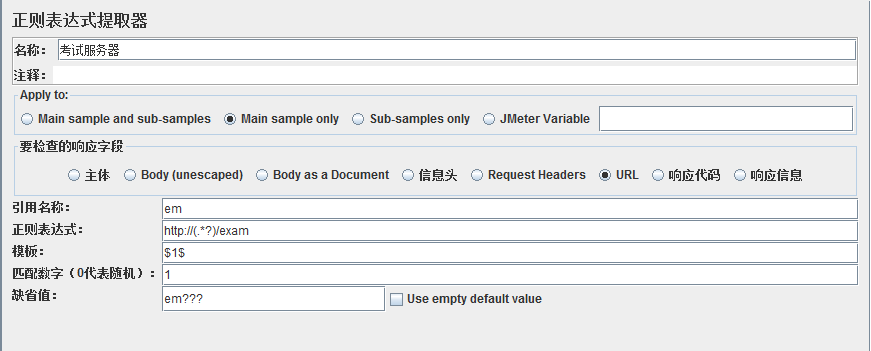




问题4：根据“获取试卷”的相应数据“http://e1.edu-edu.com

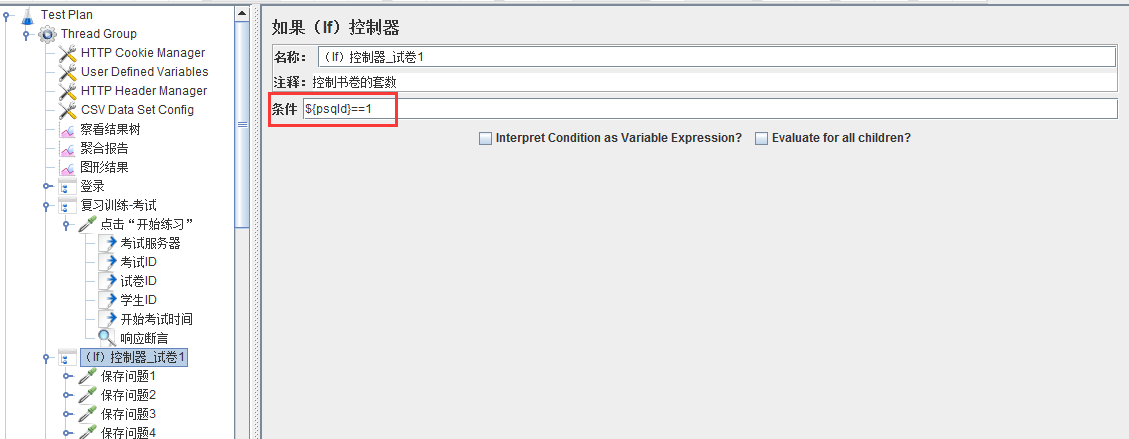
/exam/student/exam/start/1/1/1/9300005\_02680\_STEP/1467619758092?lt=10&cj=0&cc=0&co=0&m=b88d5fd288bfcd2419f155028953afda&et=1467620358092&asr=1&asa=1&ss=0&s=1”

可知“e1.edu-edu.com”、“ e2.edu-edu.com”、“ e3.edu-edu.com”是随机变化的。所以要为此服务器也要做一个正则。



问题5：此考试系统中的试卷是生成5份，然后学生随机打开某一份试卷。

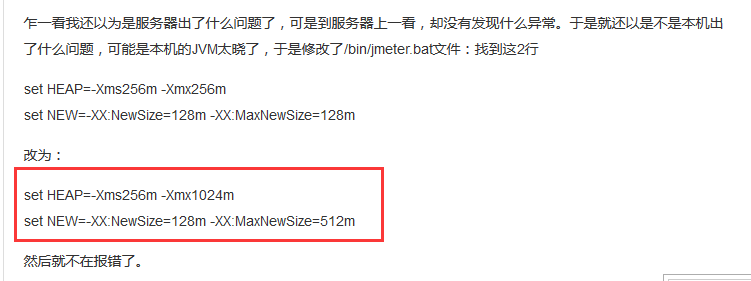
所以在这里添加一个“如果（If）控制器”，并添加“条件”。此试卷为当psqId=1时，进入第1套试卷。当然后面需要添加5套试卷，分别当psqId为相应的值时，进入此套试卷进行考试。



问题6：当线程数较大时，进行进行运行可能会出现Jmeter报错的情况。

需要txt打开Jmeter下的\apache-jmeter-3.0\apache-jmeter-3.0\bin\jmeter.bat。

修改如下内容。



问题7：Response message: Non HTTP response message: connect timed out

错误分析：通过Load time值看，由于该线程耗费时间（3002）大于设置的connect time out（3000ms），因此抛出该异常。问题可能是由于服务端有较多请求正在处理（且处理时间较长），导致JMeter不能连接上服务器而产生的。

JMeter原始错误信息：

概要：

Thread Name: 线程组 1-367

Sample Start: 2013-07-05 11:04:17 CST

Load time: 3002

Latency: 0

Size in bytes: 1677

Headers size in bytes: 0

Body size in bytes: 1677

Sample Count: 1

Error Count: 1

Response code: Non HTTP response code: java.net.SocketTimeoutException

Response message: Non HTTP response message: connect timed out

**详细信息：**

java.net.SocketTimeoutException: connect timed out

at java.net.PlainSocketImpl.socketConnect(Native Method)

at java.net.PlainSocketImpl.doConnect(PlainSocketImpl.java:351)

at java.net.PlainSocketImpl.connectToAddress(PlainSocketImpl.java:213)

at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:200)

at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:366)

at java.net.Socket.connect(Socket.java:529)

at sun.net.NetworkClient.doConnect(NetworkClient.java:158)

at sun.net.www.http.HttpClient.openServer(HttpClient.java:395)

at sun.net.www.http.HttpClient.openServer(HttpClient.java:530)

at sun.net.www.http.HttpClient.<init>(HttpClient.java:234)

at sun.net.www.http.HttpClient.New(HttpClient.java:307)

at sun.net.www.http.HttpClient.New(HttpClient.java:324)

at sun.net.www.protocol.http.HttpURLConnection.getNewHttpClient(HttpURLConnection.java:970)

at sun.net.www.protocol.http.HttpURLConnection.plainConnect(HttpURLConnection.java:911)

at sun.net.www.protocol.http.HttpURLConnection.connect(HttpURLConnection.java:836)

at org.apache.jmeter.protocol.http.sampler.HTTPJavaImpl.sample(HTTPJavaImpl.java:487)

at org.apache.jmeter.protocol.http.sampler.HTTPSamplerProxy.sample(HTTPSamplerProxy.java:62)

at org.apache.jmeter.protocol.http.sampler.HTTPSamplerBase.sample(HTTPSamplerBase.java:1088)

at org.apache.jmeter.protocol.http.sampler.HTTPSamplerBase.sample(HTTPSamplerBase.java:1077)

at org.apache.jmeter.threads.JMeterThread.process\_sampler(JMeterThread.java:428)

at org.apache.jmeter.threads.JMeterThread.run(JMeterThread.java:256)

at java.lang.Thread.run(Thread.java:662)

问题8：Response message: Non HTTP response message: Read timed out

错误分析：通过返回错误信息看，发生该错误时，JMeter已经连接上服务器，但是产生read time out。从load time（2998）看，所用时间并没有超过设定超时时间（3000），因此错误不大可能是JMeter本身产生的。一种可能是，服务器那边未处理该线程的请求，或者为保证服务能力，断掉了连接。

JMeter原始错误信息：

概要：

Thread Name: 线程组 1-10

Sample Start: 2013-07-05 11:12:45 CST

Load time: 2988

Latency: 0

Size in bytes: 2431

Headers size in bytes: 0

Body size in bytes: 2431

Sample Count: 1

Error Count: 1

Response code: Non HTTP response code: java.net.SocketTimeoutException

Response message: Non HTTP response message: Read timed out

**详细信息：**

at java.net.SocketInputStream.socketRead0(Native Method)

at java.net.SocketInputStream.read(SocketInputStream.java:129)

at java.io.BufferedInputStream.fill(BufferedInputStream.java:218)

at java.io.BufferedInputStream.read1(BufferedInputStream.java:258)

at java.io.BufferedInputStream.read(BufferedInputStream.java:317)

at sun.net.www.http.HttpClient.parseHTTPHeader(HttpClient.java:697)

at sun.net.www.http.HttpClient.parseHTTP(HttpClient.java:640)

at sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConnection.java:1195)

at sun.net.www.protocol.http.HttpURLConnection.getHeaderField(HttpURLConnection.java:2300)

at java.net.URLConnection.getHeaderFieldInt(URLConnection.java:579)

at java.net.URLConnection.getContentLength(URLConnection.java:474)

at org.apache.jmeter.protocol.http.sampler.HTTPJavaImpl.readResponse(HTTPJavaImpl.java:230)