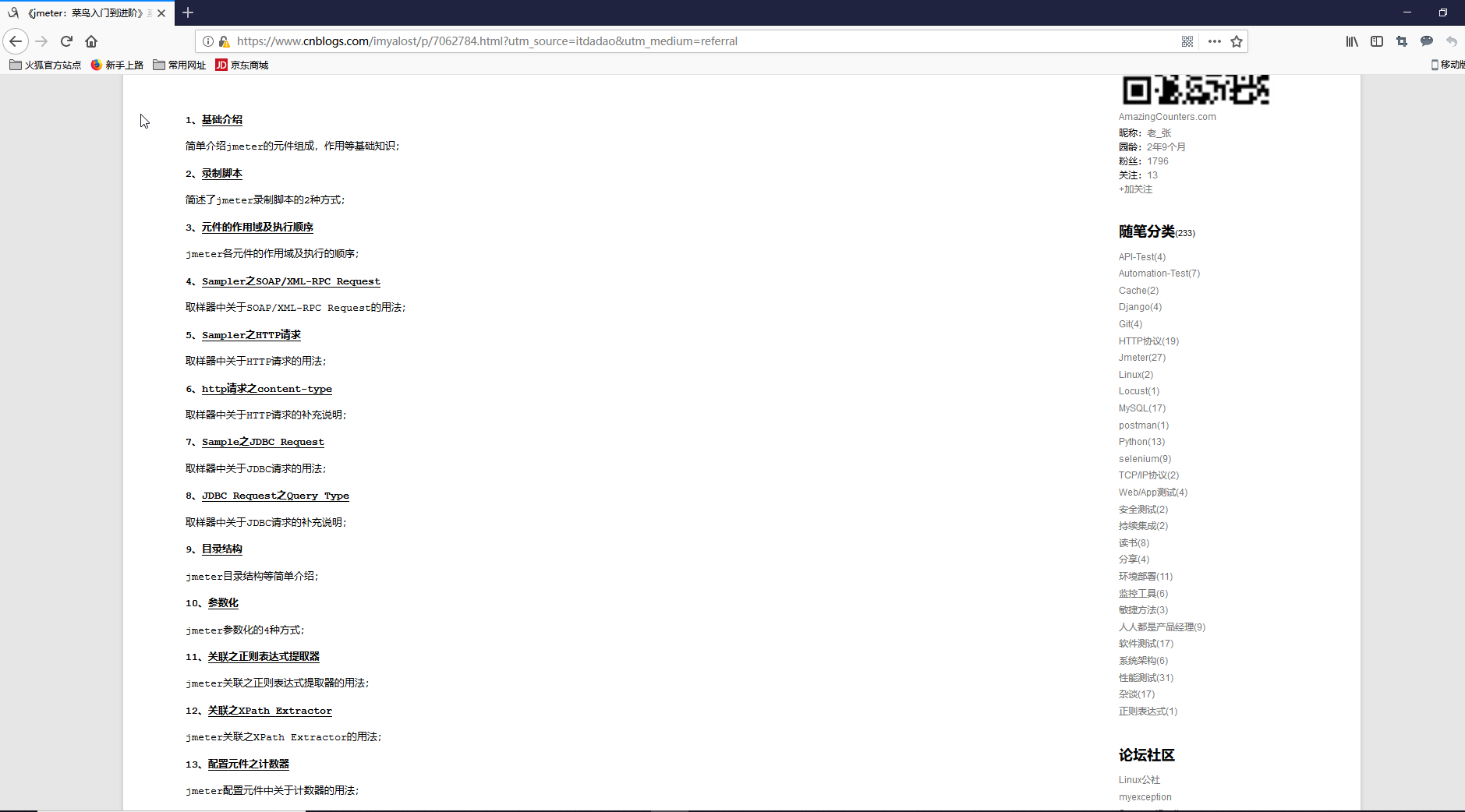
# 基础介绍：

## 1.1、模块介绍

各个模块和功能介绍这里不再罗列赘述，详见

<https://www.cnblogs.com/imyalost/p/7062784.html?utm_source=itdadao&utm_medium=referral>



## 1.2、目录介绍

### 1.bin目录

examples:目录中有CSV样例

jmeter.bat 　　　　　　　 windows的启动文件

jmeter.log 　　　　　　　  jmeter运行日志文件

jmeter.sh 　　　　　　　   linux的启动文件

jmeter.properties 　　　  系统配置文件

jmeter-server.bat            windows分布式测试要用到的服务器配置

jmeters-server                linux分布式测试要用的服务器配置

其中系统配置文件中的SSL设置重点关注如下几个：

# 指定HTTPS协议层

https.default.protocol=TLS

# 指定SSL版本  
https.default.protocol=SSLv3

# 设置启动的协议  
https.socket.protocols=SSLv2Hello SSLv3 TLSv1

# 缓存控制，控制SSL是否可以在多个迭代中重用  
https.use.cached.ssl.context=true

### 2.docs目录

　　接口文档目录。例C:\apache-jmeter-3.0\docs\api下的index.html

### 3.extras目录

　　扩展插件目录。提供了对Ant的支持，可以使用Ant来实现自动化测试，例如批量脚本执行，产生html格式的报表，测试运行时，可以把测试数据记录下来，jmeter会自动生成一个.jtl文件，将该文件放到extras目录下，运行"ant -Dtest=文件名 report"，就可以生成测试统计报表。

### 4.lib目录

　　所用到的插件目录，里面均为jar包。jmeter会自动在jmeter\_HOME/lib和ext目录下寻找需要的类，lib下存放JMeter所依赖的外部jar：如httpclient.jar、httpcore.jar、httpmime.jar等等。

　　其中lib\ext目录下存放有Jmeter依赖的核心jar包，ApacheJMeter\_core.jar、ApacheJMeter\_java.jar在写client端需要引用，JMeter插件包也在此目录下。

　　lib\junit下存放junit测试脚本.

### 5.Licenses目录

　　jmeter证书目录

### 6.Printable\_docs目录

用户使用手册，例C:\apache-jmeter-3.0\printable\_docs下的index.html

# 二、工具使用：

## 2.1 JDK安装

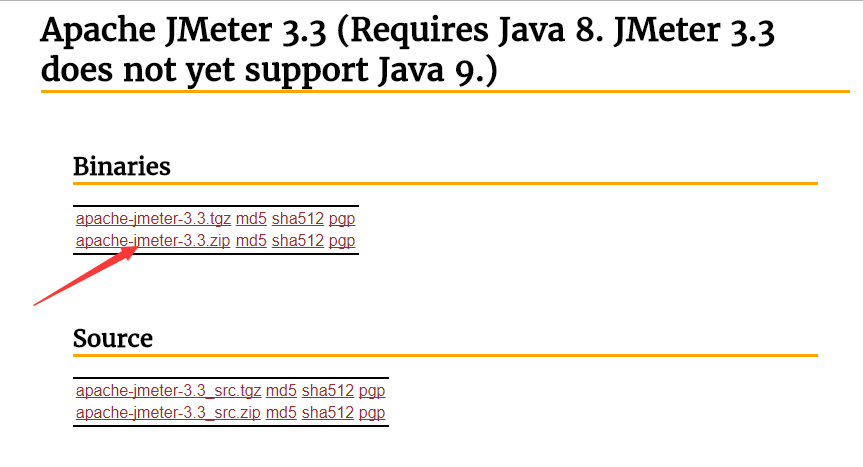
由于Jmeter是基于java开发，首先需要下载安装JDK （目前JMeter只支持到Java 8，尚不支持 Java 9）

1. 官网下载地址：[http://www.oracle.com/technetwork/java/javase/downloads/index.html](http://www.oracle.com/technetwork/java/javase/downloads/index.html" \t "_blank)   
2. 选择Java SE 8u151/ 8u152，点击JDK下载   
3. 安装下载的JDK   
4. 配置系统环境变量

## 2.2 JMeter安装

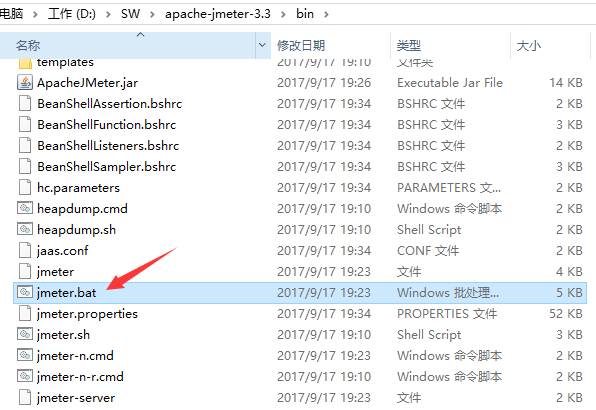
1.官网下载地址：[http://jmeter.apache.org/download\_jmeter.cgi](http://jmeter.apache.org/download_jmeter.cgi" \t "_blank)

2.下载最新JMeter 3.3版本：apache-jmeter-3.3.zip



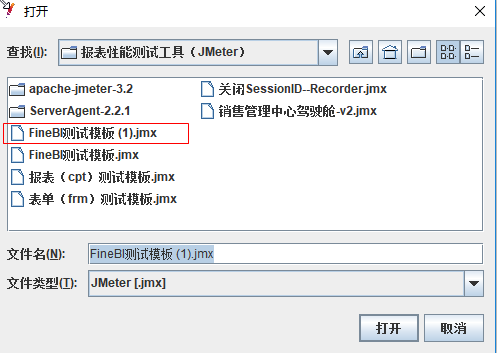
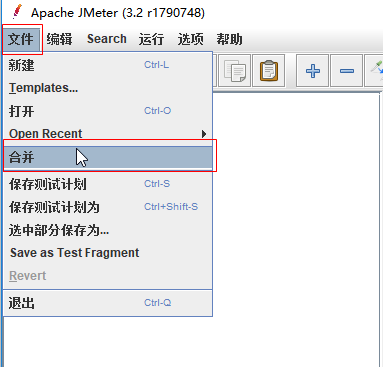
3.下载完成后解压zip包

4.启动JMeter   
双击JMeter解压路径（apache-jmeter-3.3\bin）bin下面的jmeter.bat即可

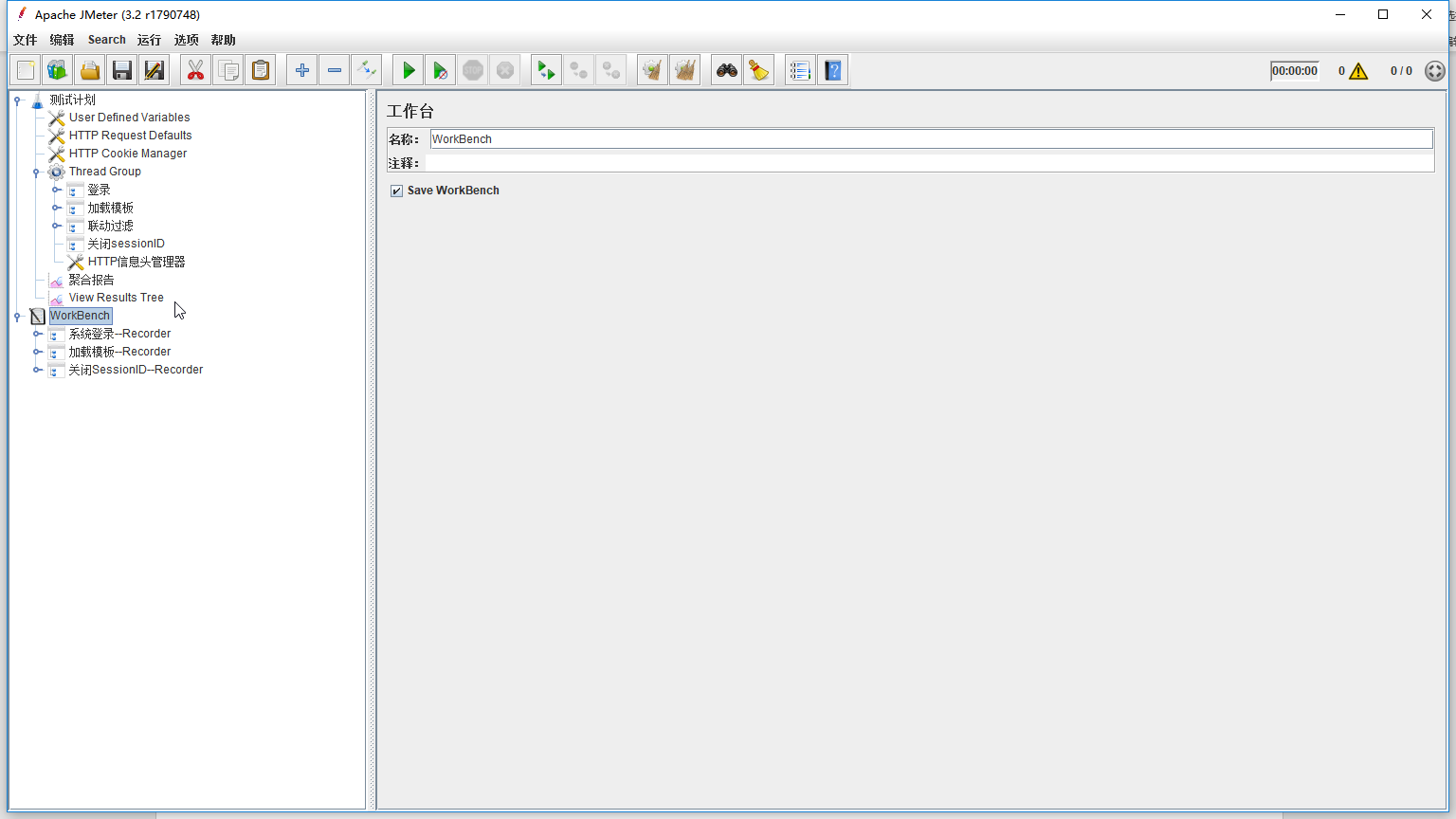


## 2.3启动和引入模版

打开界面, 文件-合并-选择"测试模版"-确定,

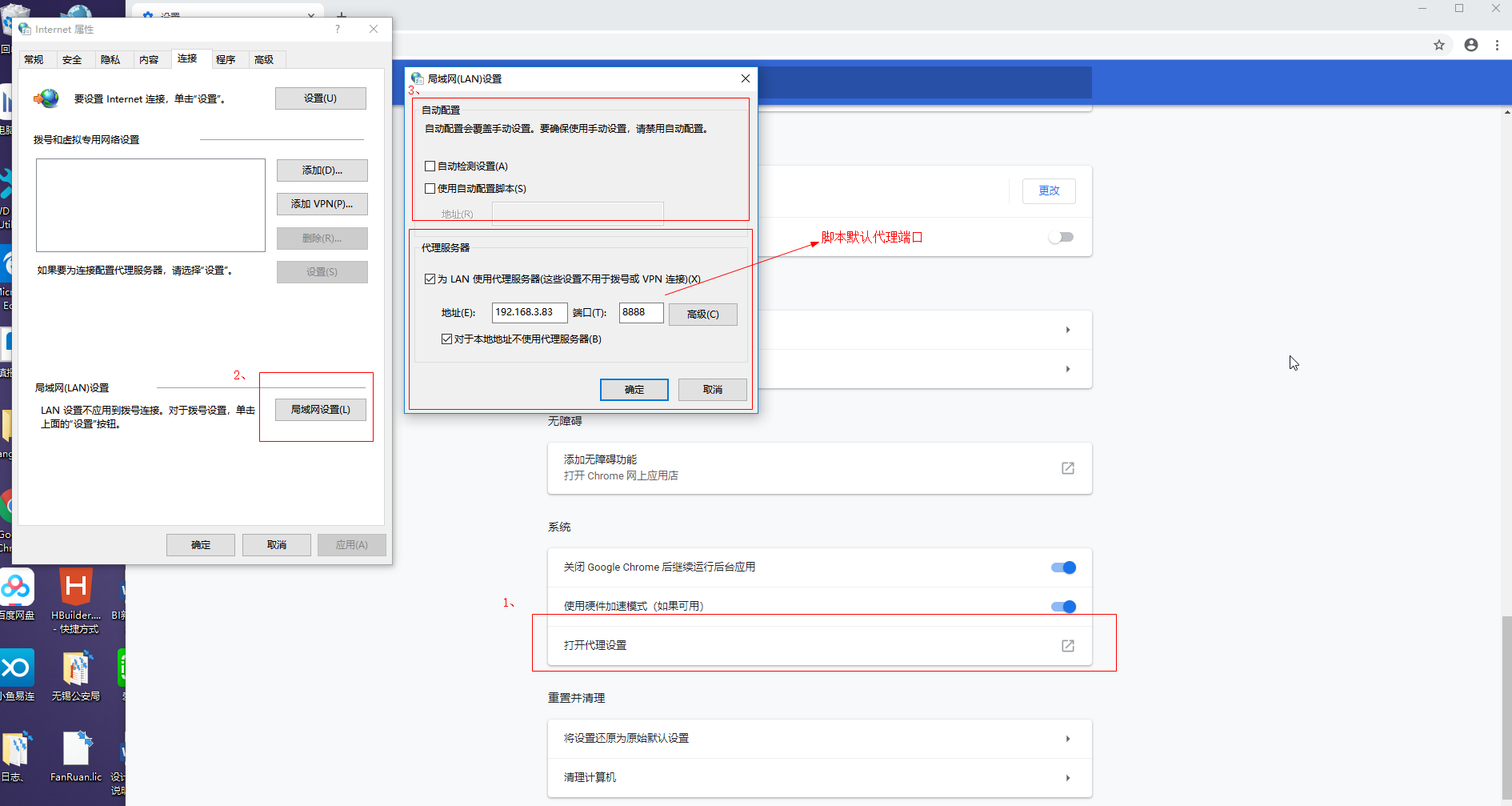


结果如下：



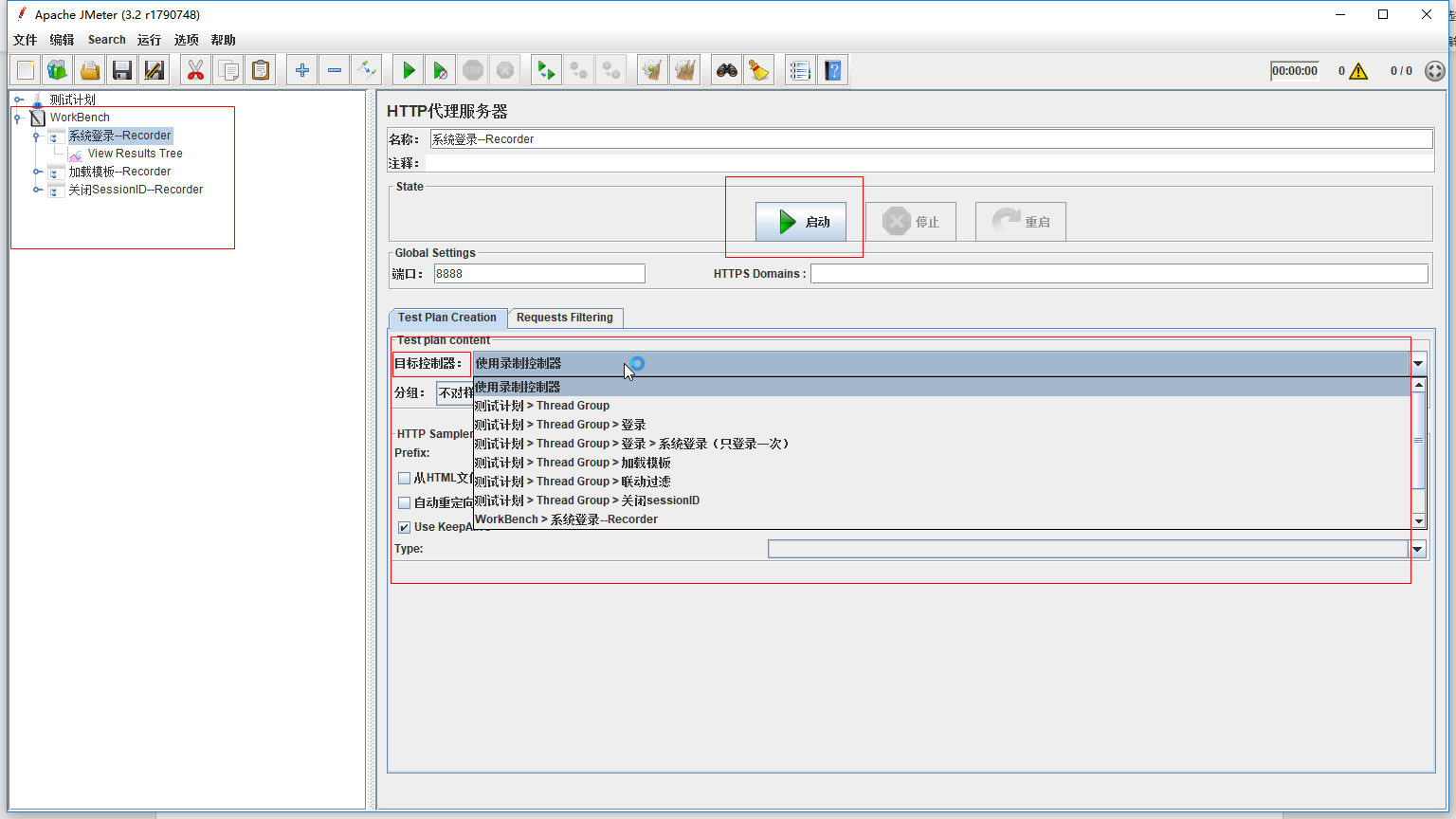
## 2.4**利用jmeter代理服务器进行脚本录制**

#### 1、配置浏览器代理（谷歌为例）



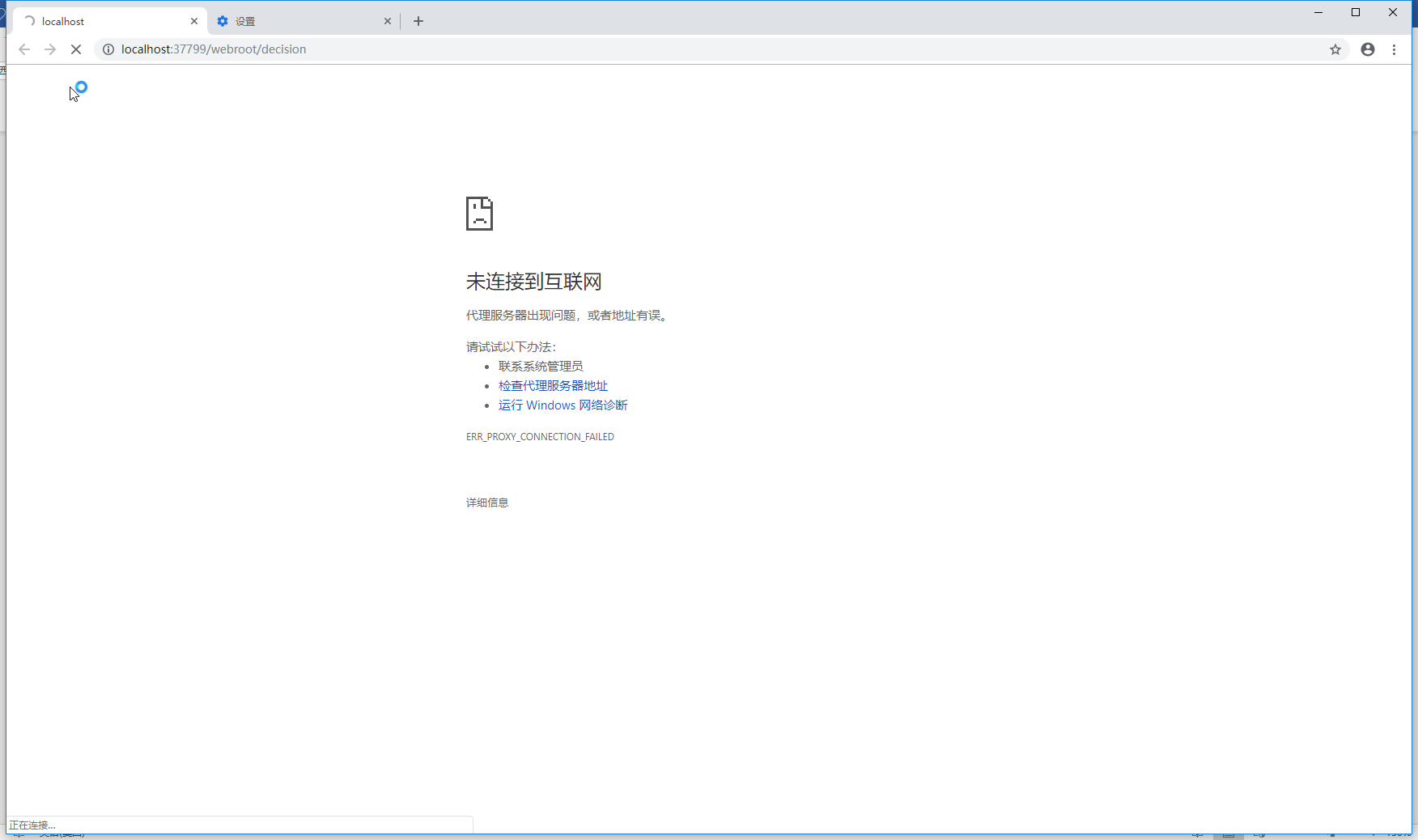
#### 2、录制

选择启动器——选择事务控制器——启动录制——浏览器端进行事物操作——操作结束后回到Jmeter点击停止，完成一个事务的录制。



PS：出现如下现象，证明代理生效

1. 浏览器代理设置完之后

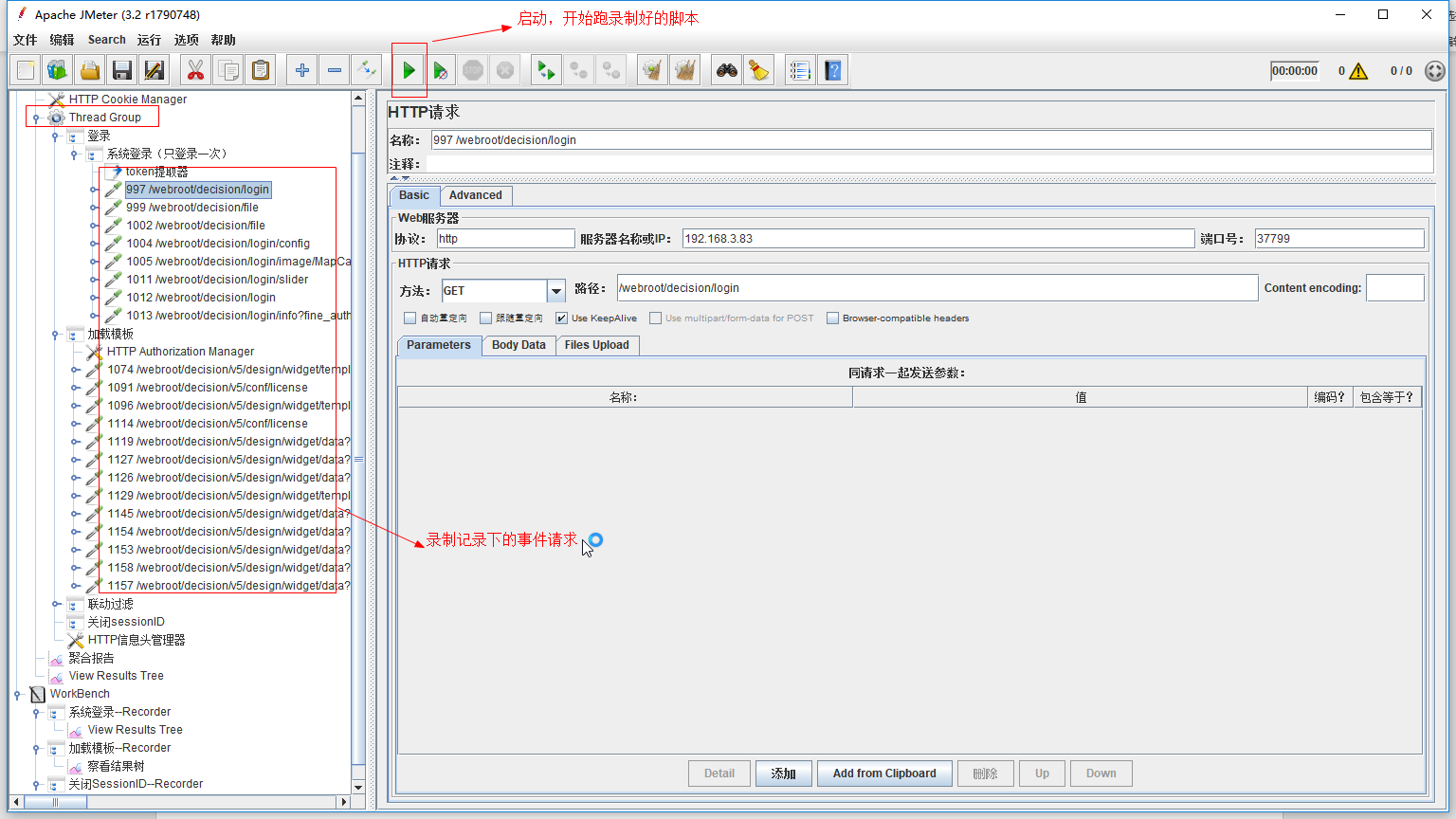


1. 点击启动录制后，可正常访问

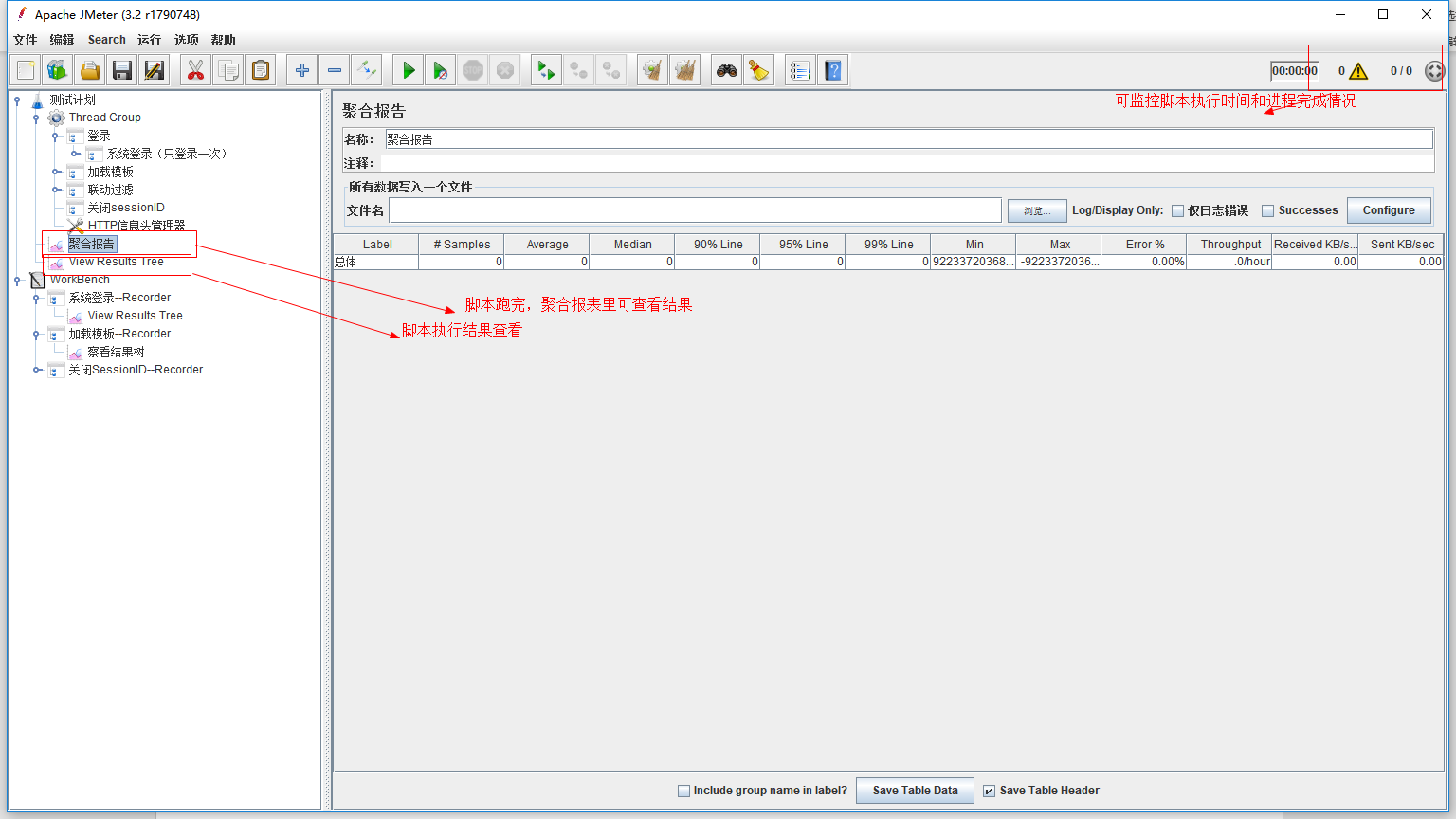
## 2.5跑脚本，查看压测结果

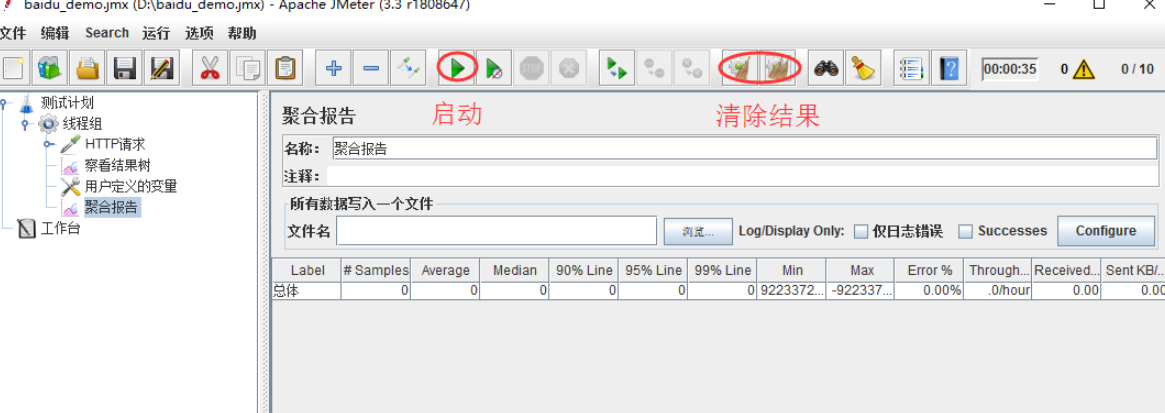
### 1、跑录制的脚本

完成上述录制后，在Thread Group里可以看到录制的HTTP请求，点击工具栏的启动按钮，开始跑脚本。

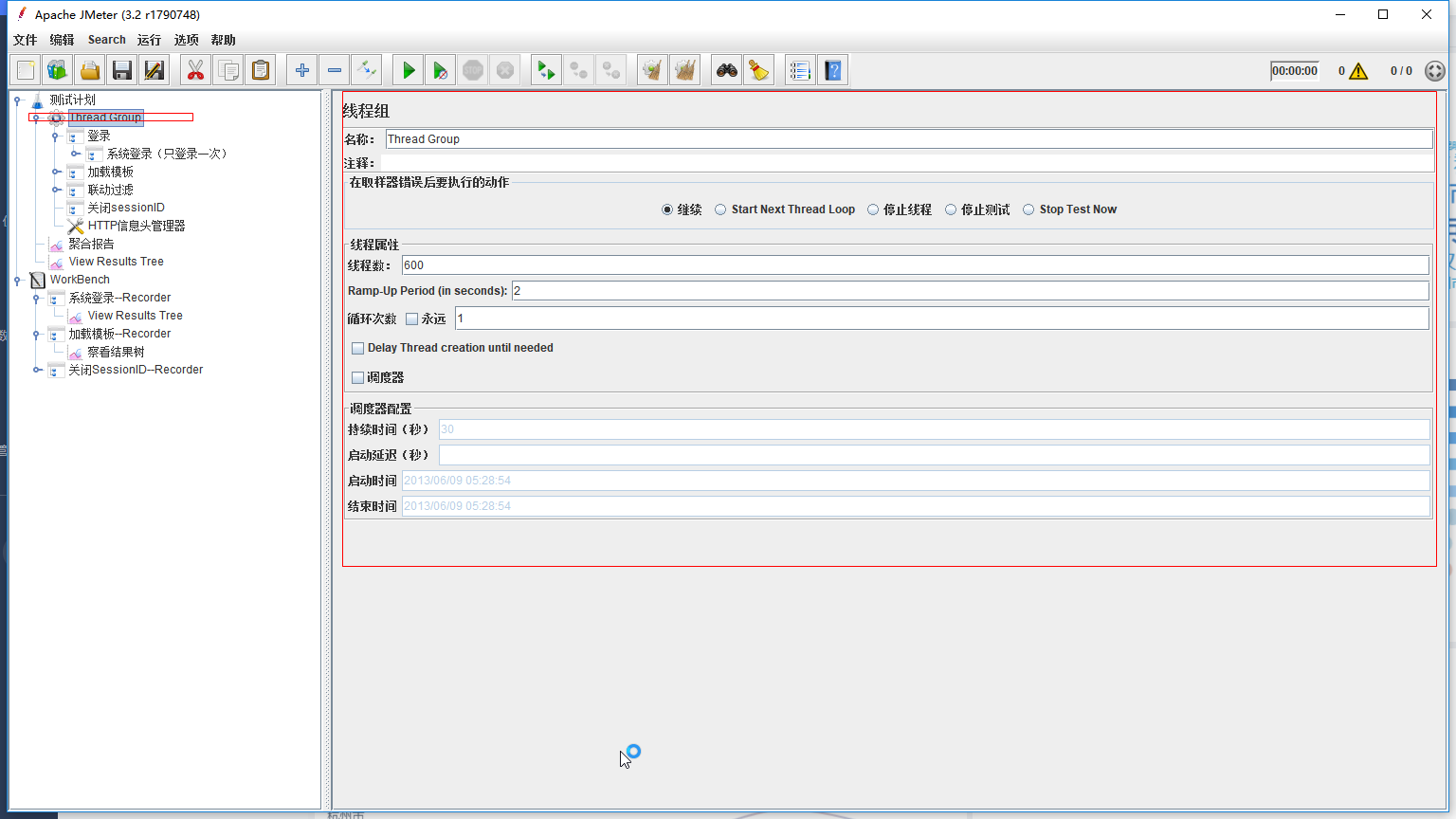


### 2、结果查看





### 3、脚本参数配置

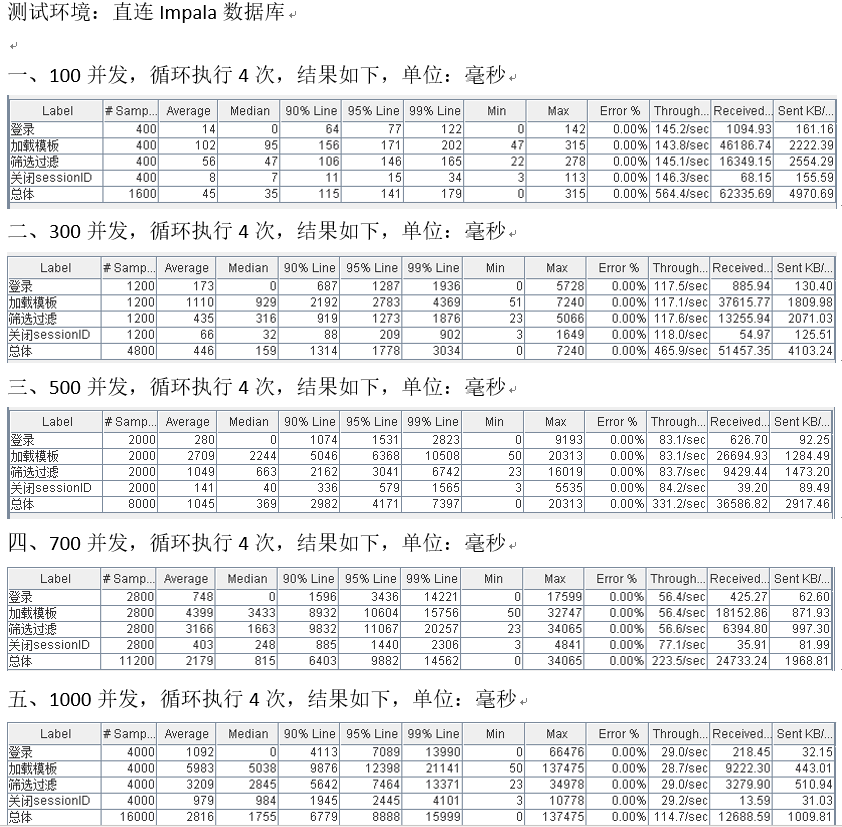


### 4、线程组参数详解

1. 线程数：虚拟用户数。一个虚拟用户占用一个进程或线程。设置多少虚拟用户数在这里也就是设置多少个线程数。   
2. Ramp-Up Period(in seconds)准备时长：设置的虚拟用户数需要多长时间全部启动。如果线程数为10，准备时长为2，那么需要2秒钟启动10个线程，也就是每秒钟启动5个线程。   
3. 循环次数：每个线程发送请求的次数。如果线程数为10，循环次数为100，那么每个线程发送100次请求。总请求数为10\*100=1000 。如果勾选了“永远”，那么所有线程会一直发送请求，一到选择停止运行脚本。   
4. Delay Thread creation until needed：直到需要时延迟线程的创建。   
5. 调度器：设置线程组启动的开始时间和结束时间(配置调度器时，需要勾选循环次数为永远)   
持续时间（秒）：测试持续时间，会覆盖结束时间   
启动延迟（秒）：测试延迟启动时间，会覆盖启动时间   
启动时间：测试启动时间，启动延迟会覆盖它。当启动时间已过，手动只需测试时当前时间也会覆盖它。   
结束时间：测试结束时间，持续时间会覆盖它。

因为接口调试需要，我们暂时均使用默认设置，待后面真正执行性能测试时再回来配置。

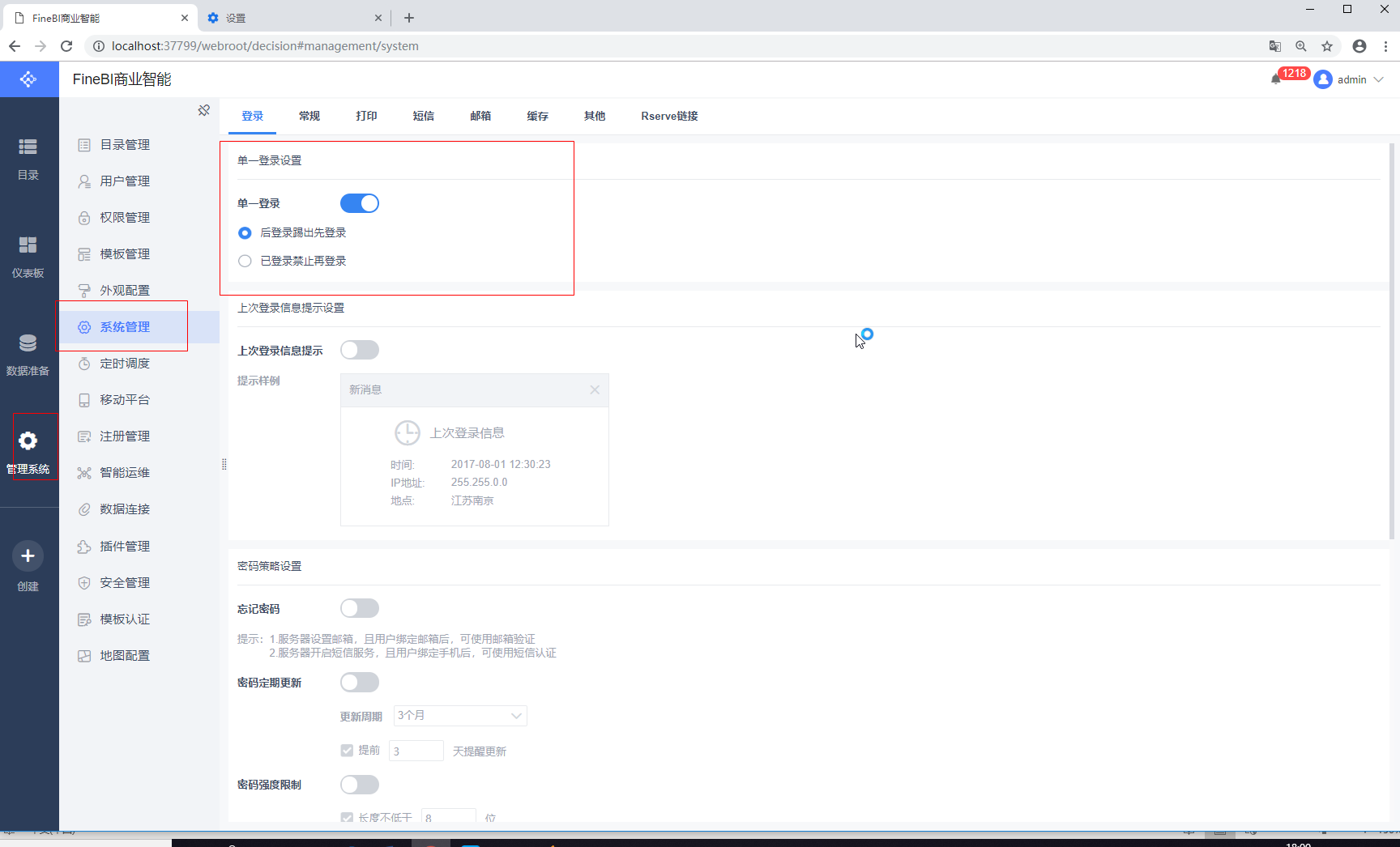
PS：分享一组测试报告



# 三、注意事项（坑点）

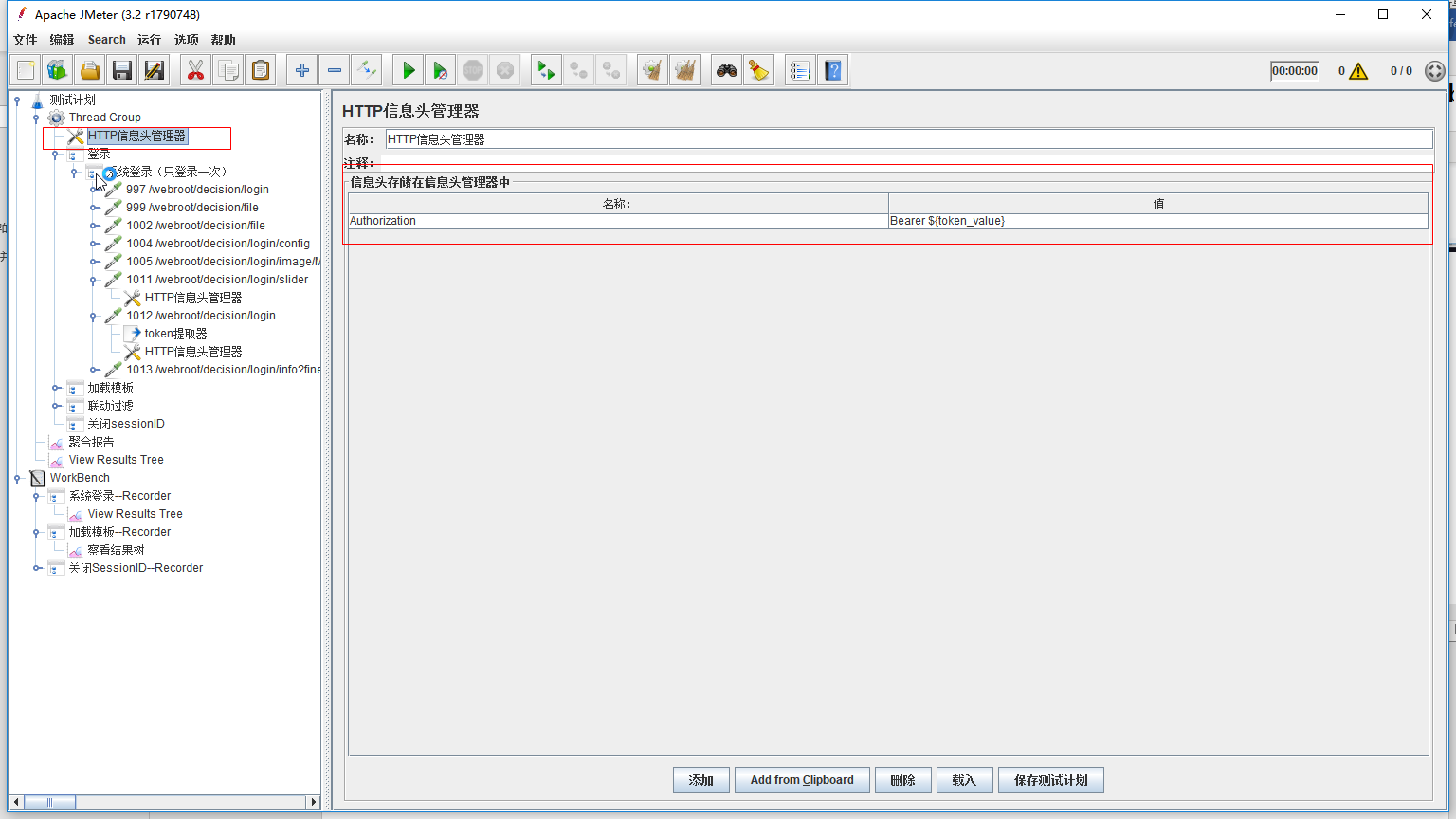
## 1、关闭单一登陆限制

（坑点就是目前全功能lic文件不支持关闭单一登陆，需要单独申请）

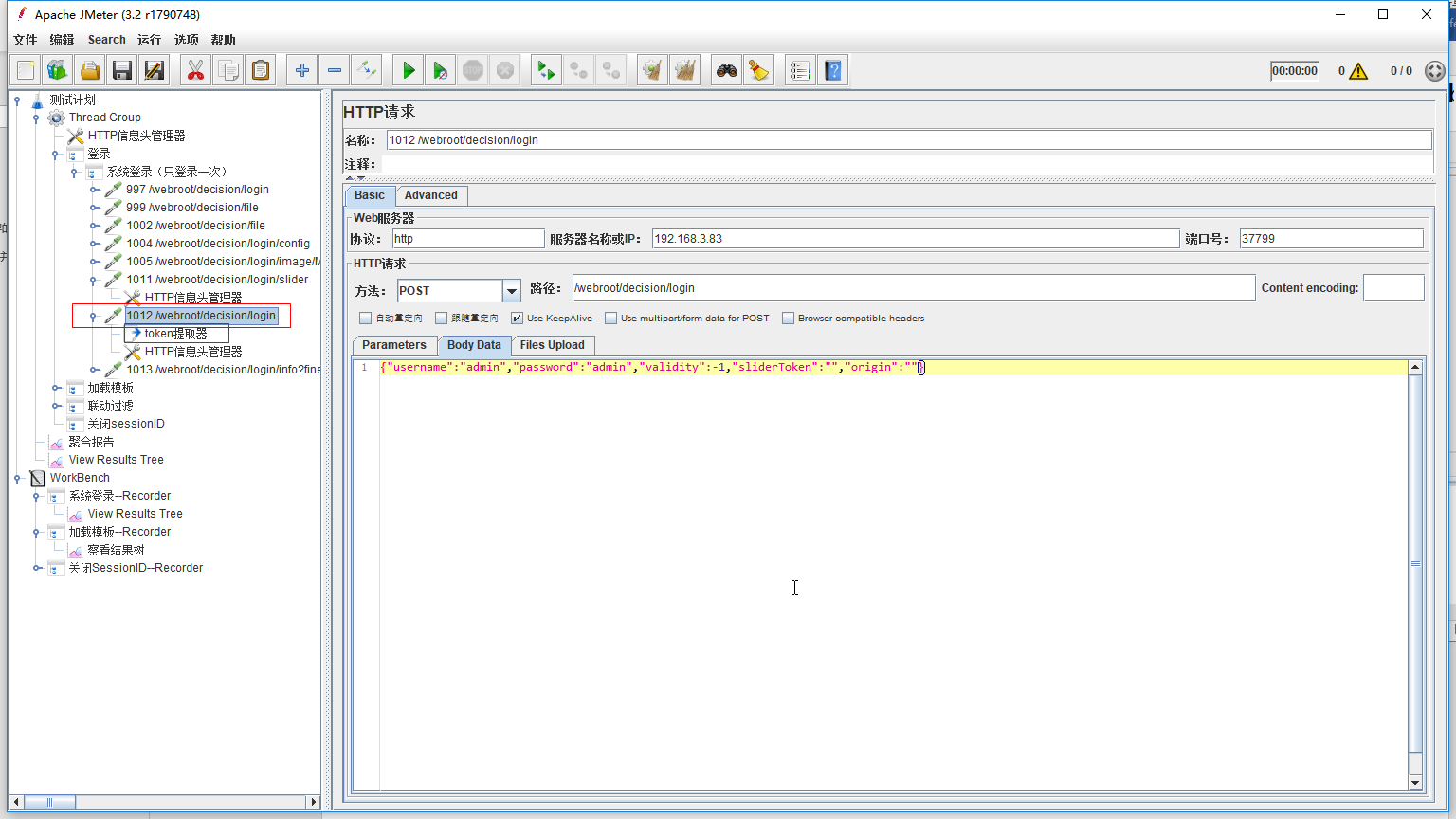


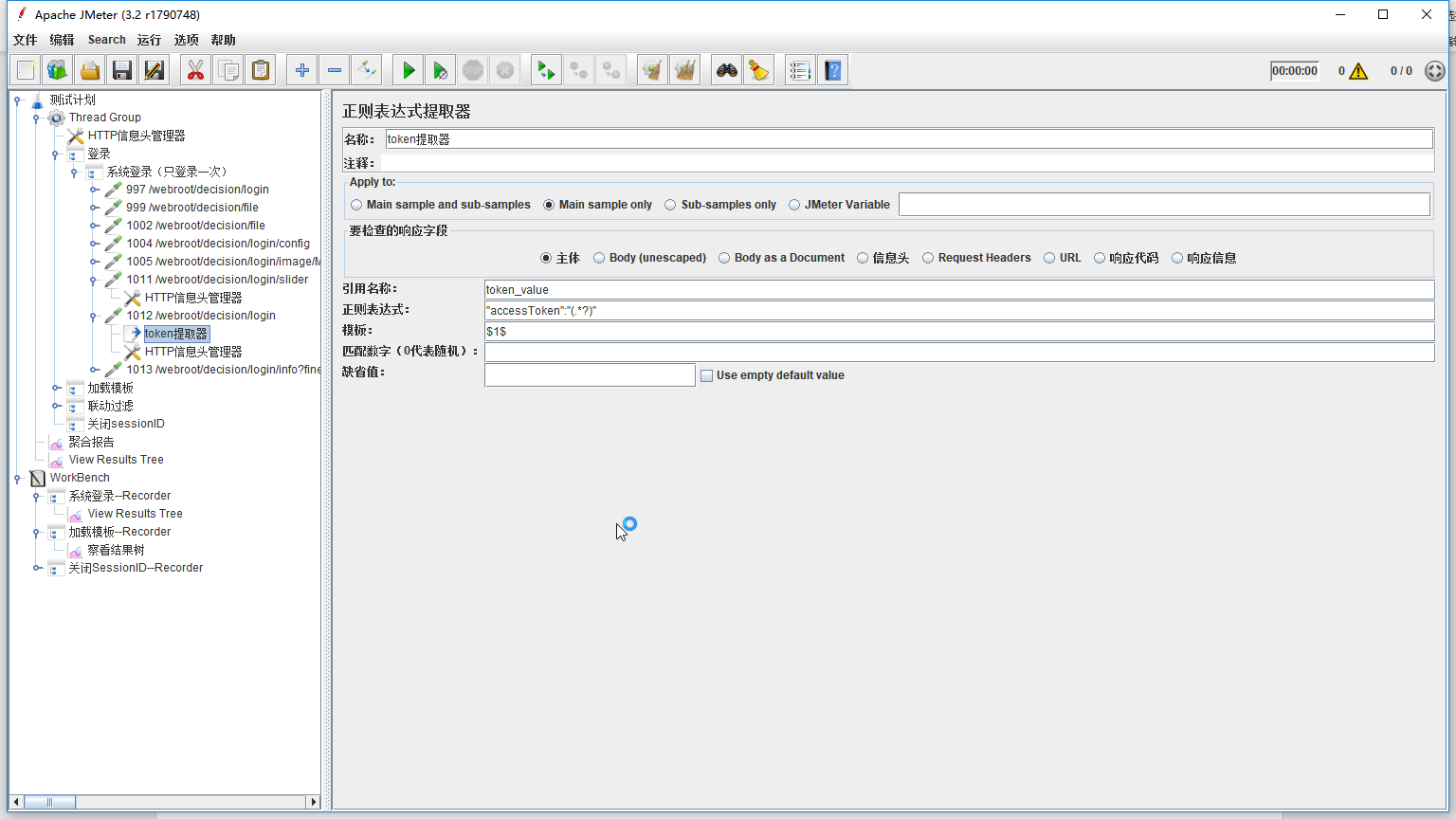
## 2、脚本执行报错（结果树登陆请求，响应数据“data”false）

（1）在事务控制器前加一个HTTP信息头管理器

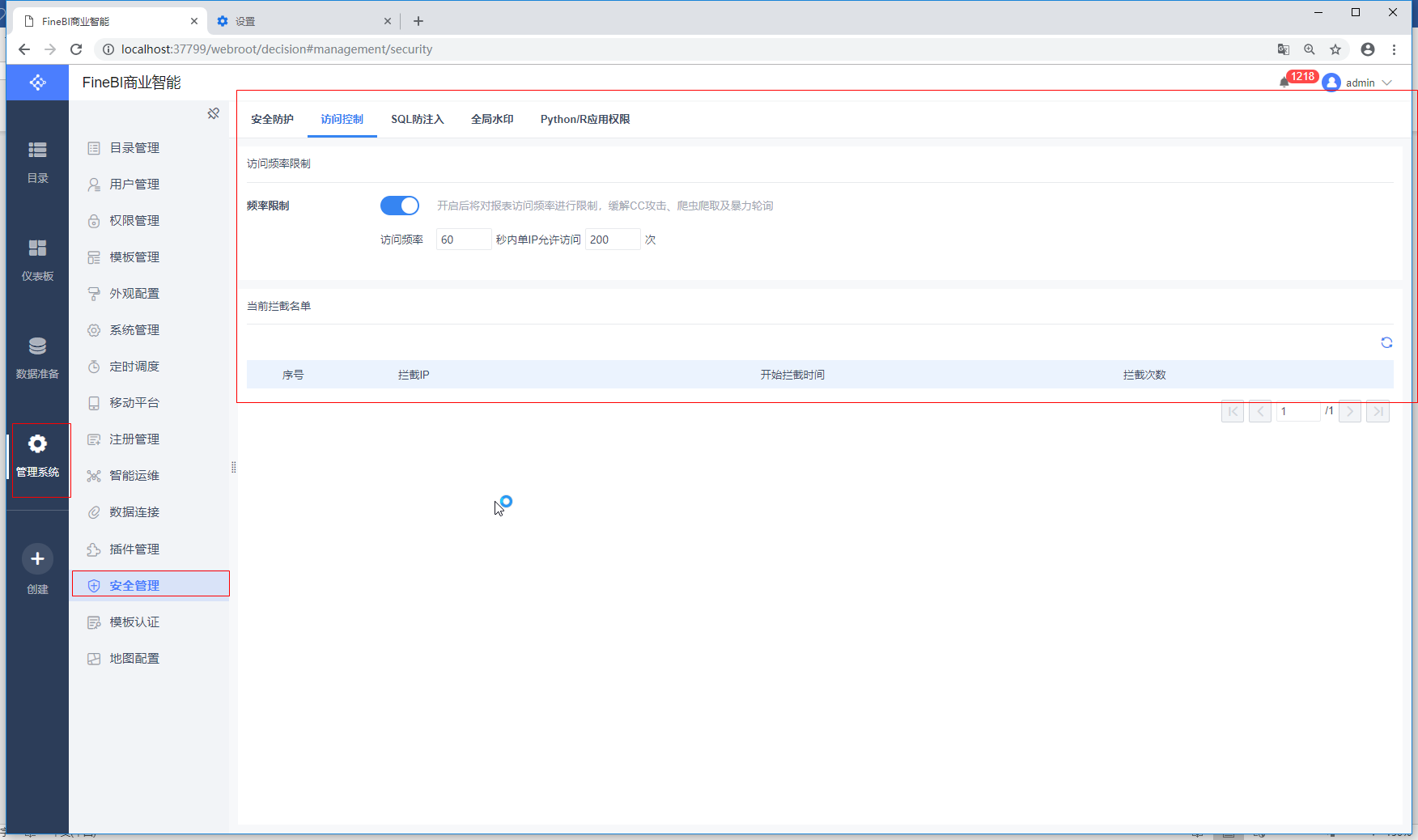


（2）在登陆请求里加一个token提取器





## 3、访问频率限制关闭



## 4、测试结果解释

1、Label - 请求对应的name属性值。

2、 Samples - 具有相同标号的样本数，总的发出请求数。

3、Average - 请求的平均响应时间。

4、Median - 50%的样本都没有超过这个时间。这个值是指把所有数据按由小到大将其排列，就是排列在第50%的值。

5、90% Line - 90%的样本都没有超过这个时间。这个值是指把所有数据按由小到大将其排列，就是排列在第90%的值。

6、95% Line - 95%的样本都没有超过这个时间。这个值是指把所有数据按由小到大将其排列，就是排列在第95%的值。

7、99% Line - 99%的样本都没有超过这个时间。这个值是指把所有数据按由小到大将其排列，就是排列在第99%的值。

8、Min - 最小响应时间。

9、Max - 最大响应时间。

10、Error % - 本次测试中，有错误请求的百分比。

11、Throughput - 吞吐量是以每秒/分钟/小时的请求量来度量的。这里表示每秒完成的请求数。

12、Received KB/sec - 收到的千字节每秒的吞吐量测试。

13、Sent KB/sec - 发送的千字节每秒的吞吐量测试。

## 5、结果图形分析

下载TPS图插件，放在lib文件夹下即可。

## C:\Users\Liangxiaofeng\Documents\WXWork\1688853806313224\Cache\Image\2019-03\企业微信截图_15534323718889.png