**Jmeter接口测试总结**

## Jmeter安装

安装包准备：

**apache-jmeter-2.13.zip**

下载地址：<http://jmeter.apache.org/download_jmeter.cgi>

**jdk-8u73-windows-x64.exe**

下载地址：<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

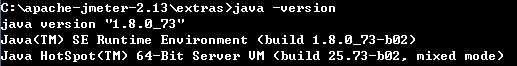
安装步骤

1. 安装jdk，点安装文件直接下一步，默认路径安装
2. 配置环境变量

新建系统变量：JAVA\_HOME C:\Program Files\Java\jdk1.8.0\_73

系统变量PATH 添加 %JAVA\_HOME%\bin;

运行java命令 查看是否配置成功



1. 解压jmeter压缩包
2. 配置jmeter环境变量

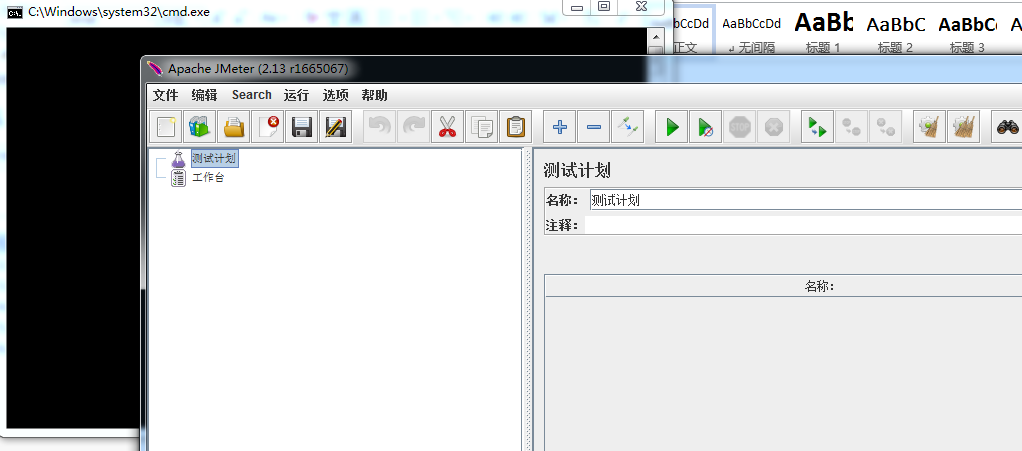
新建系统变量：JMETER\_HOME C:\apache-jmeter-2.13

系统变量 PATH 添加 %JMETER\_HOME%;

系统变量 CLASSPATH 添加

%JMETER\_HOME%\lib\ext\ApacheJMeter\_core.jar;%JMETER\_HOME%\lib\jorphan.jar;%JMETER\_HOME%\lib\logkit-1.2.jar;C:\Program Files\Java\jdk1.8.0\_73\lib\dt.jar;

Bin目录下打开Jmeter.bat文件 看是否能打开图形化界面



## ant安装

Apache Ant,是一个将软件编译、测试、部署等步骤联系在一起加以自动化的一个工具，大多用于Java环境中的软件开发。

安装包准备：

apache-ant-1.9.7-bin.zip

下载地址：<http://ant.apache.org/bindownload.cgi>

安装步骤

1. 解压ant压缩包
2. 配置ant环境变量

添加系统变量 ANT\_HOME C:\Program Files\apache-ant-1.9.7\bin

系统变量 PATH 添加 %ANT\_HOME%\bin;

运行命令查看 ant 配置是否成功

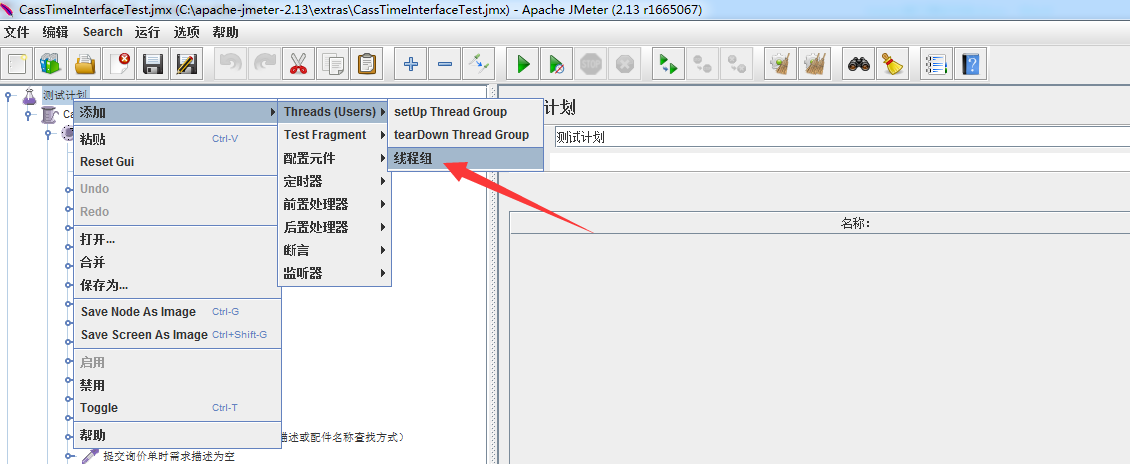


## jmeter http取样范例简析

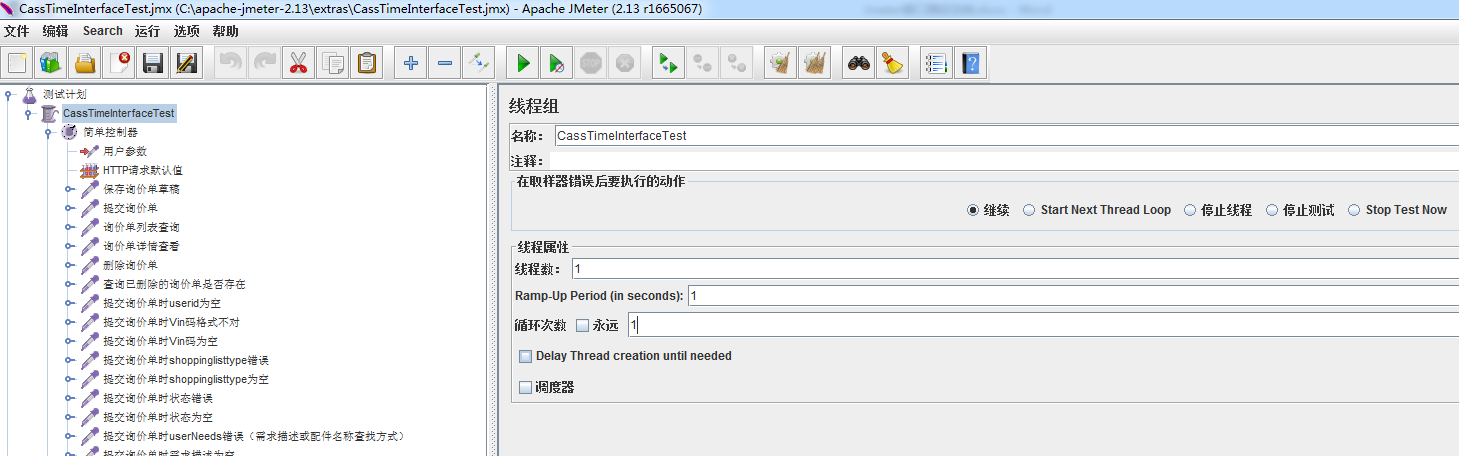
在取样之前我们先了解一下我们接口测试中要使用的一些配置原件

1、线程组

打开jmeter图形化界面，左边选中测试计划右击添加线程组，每个线程会作为一个整体执行测试计划并完全独立于他测试线程。多线程用来模拟到达服务器程序的同步连接。



Ok，这里我们重点理解一下添加线程组后其中几个重要的参数



名称肯定是写一个有意义的咯，方便知道自己是在干什么，然后就是线程组属性

线程数：每个线程会作为一个整体执行测试计划并完全独立于他测试线程。

Ramp-up Period (in seconds)：表示在这时间内创建完所有的线程

循环次数：每个时间段内线程下任务的执行次数

例如：线程数设置2，Ramp-up Period设置10，循环次数设置3，线程组内有一个提交

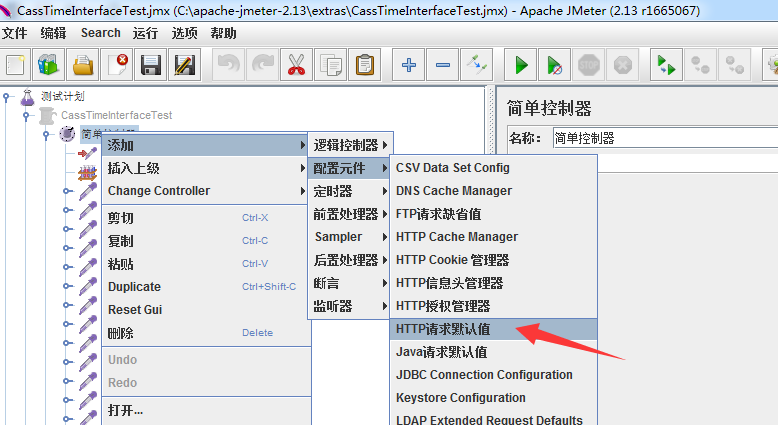
订单取样。执行测试计划 开始启动第1个线程，在第一个5s执行了3次提交，间隔5s后

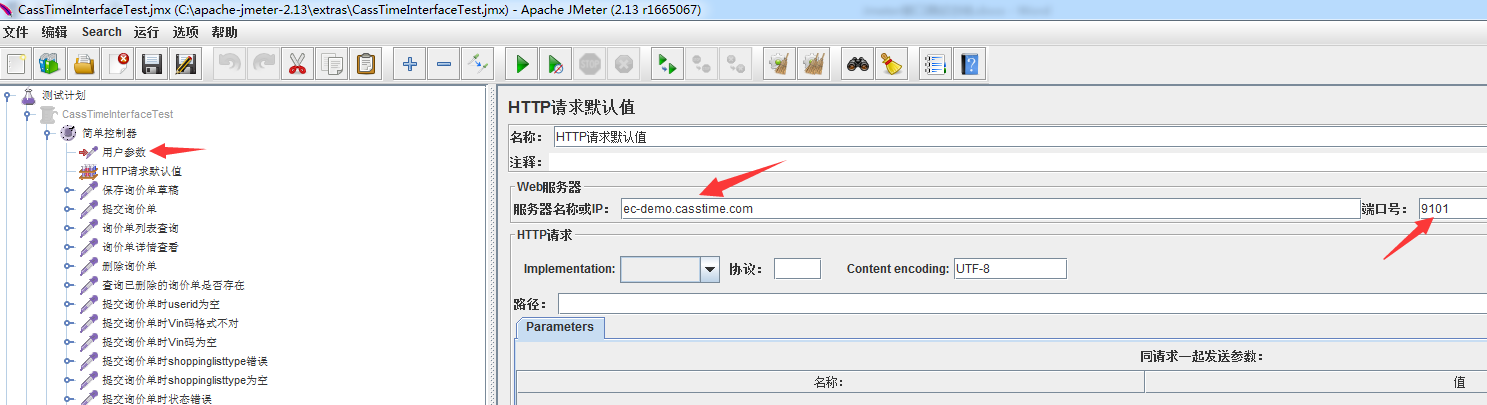
启动第2个线程 再提交3次，在10s启动完成全部的提交动作。

取样器错误后执行的动作：这个按字面意思理解就行。

2、添加http请求默认值

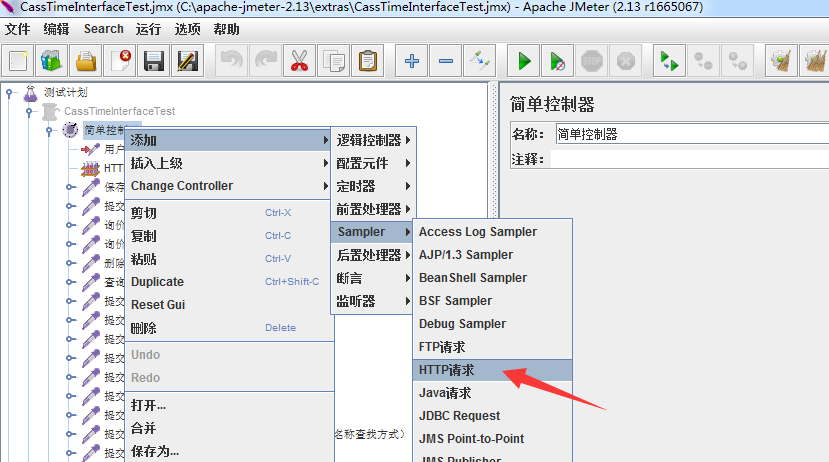
如果后面添加的取样器或者录制的脚本都是同一个域名或IP，可以在添加这个配置元件



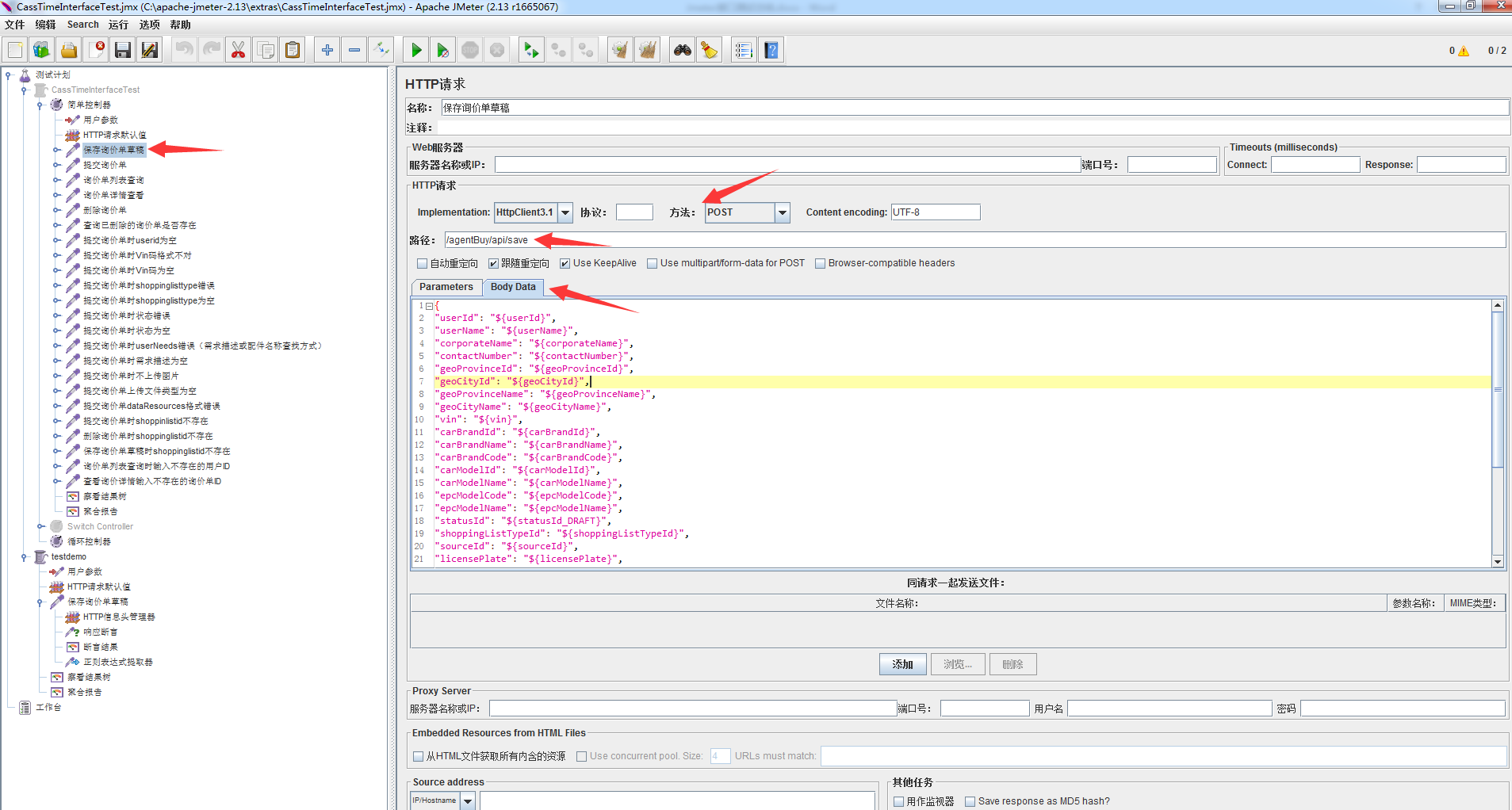


3、添加取样器

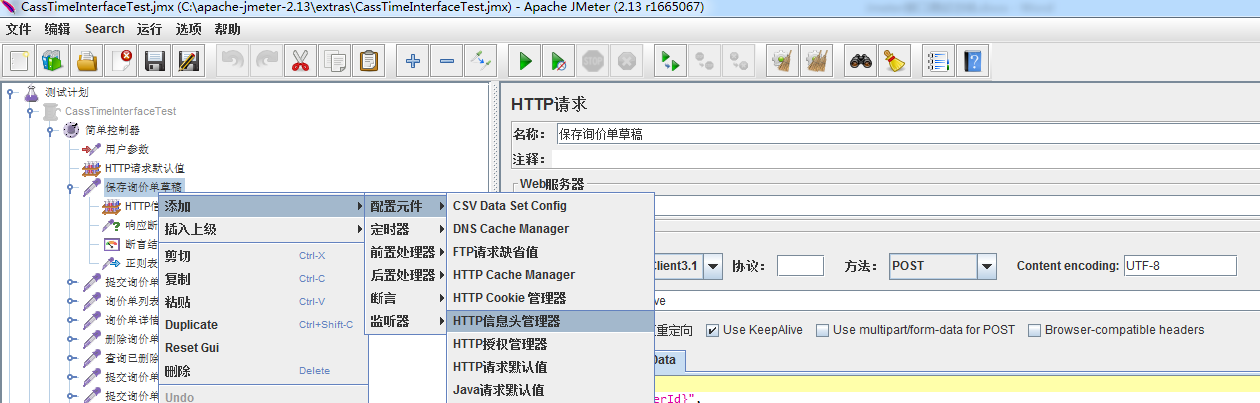
这里要注意的是根据你发送不同请求类型来添加不同取样器

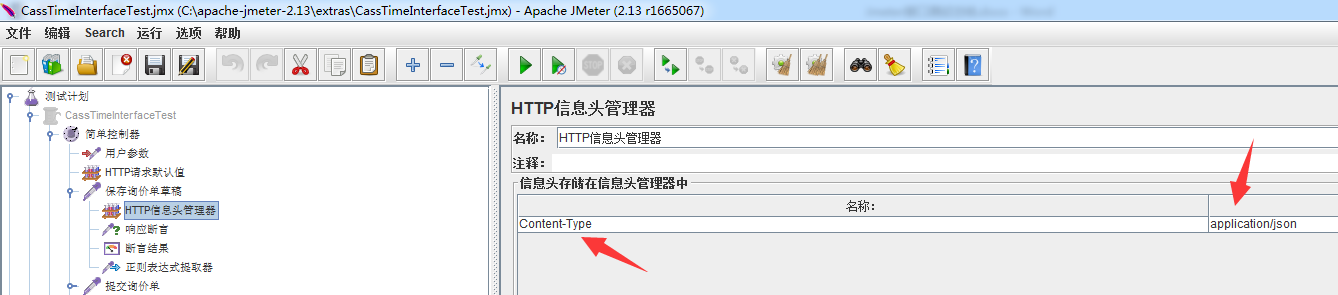


下图我们要注意几个地方第一个是发送请求的方法，是post还是get，其次发送请求的路径，然后是请求内容，这里要注意一下请求内容的格式，这里以json格式内容举例

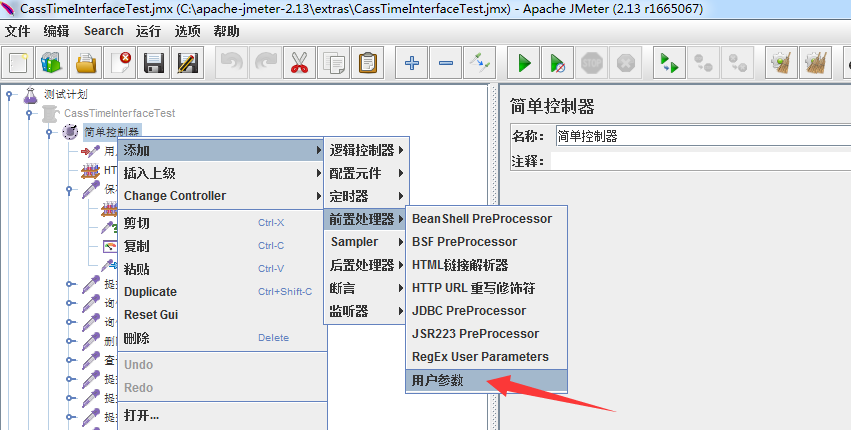


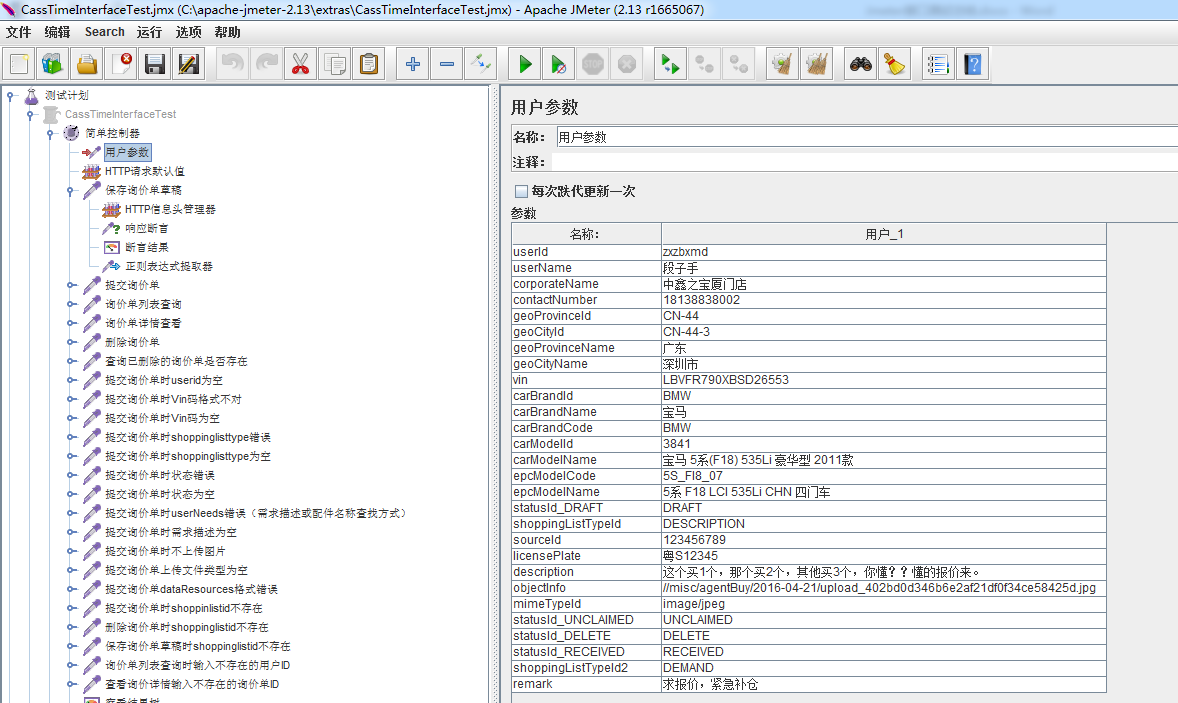
需要添加一个消息头管理器来定义发送消息内容的格式





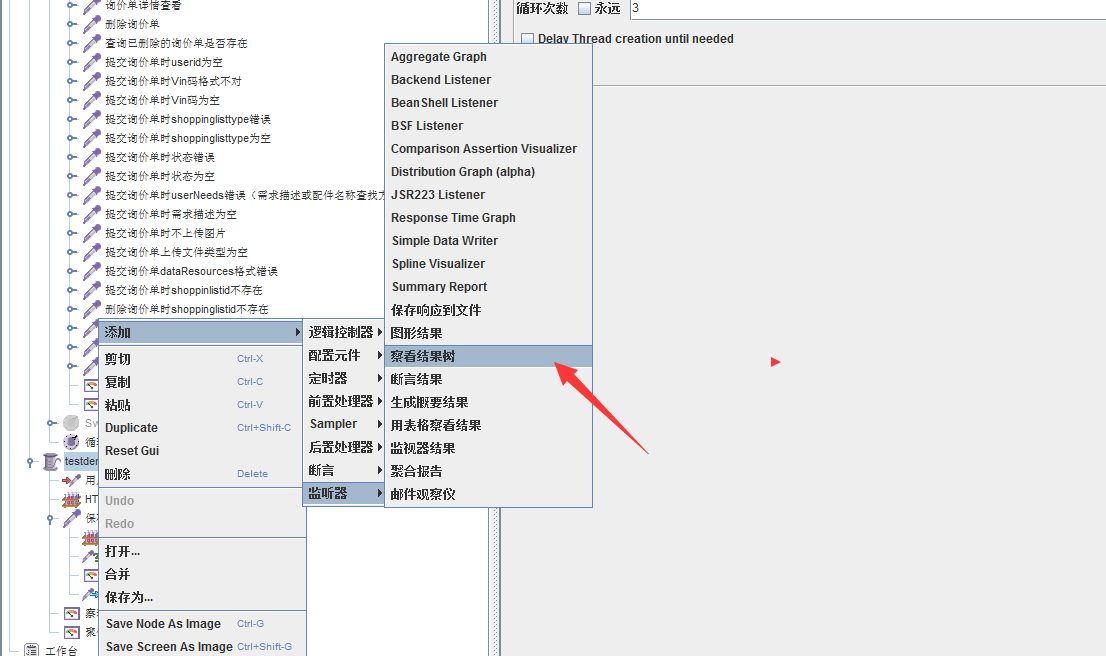
如果发送的消息参数内容很多，且需要发送多次不同的值，可以定义用户参数

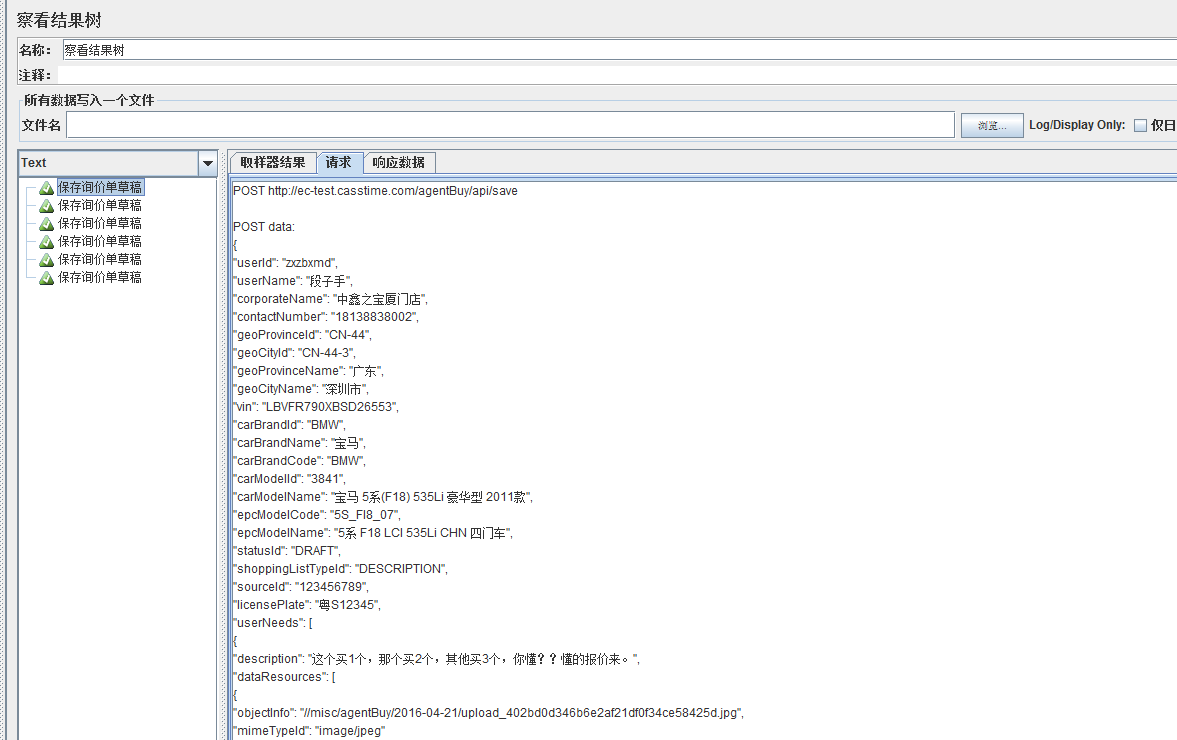




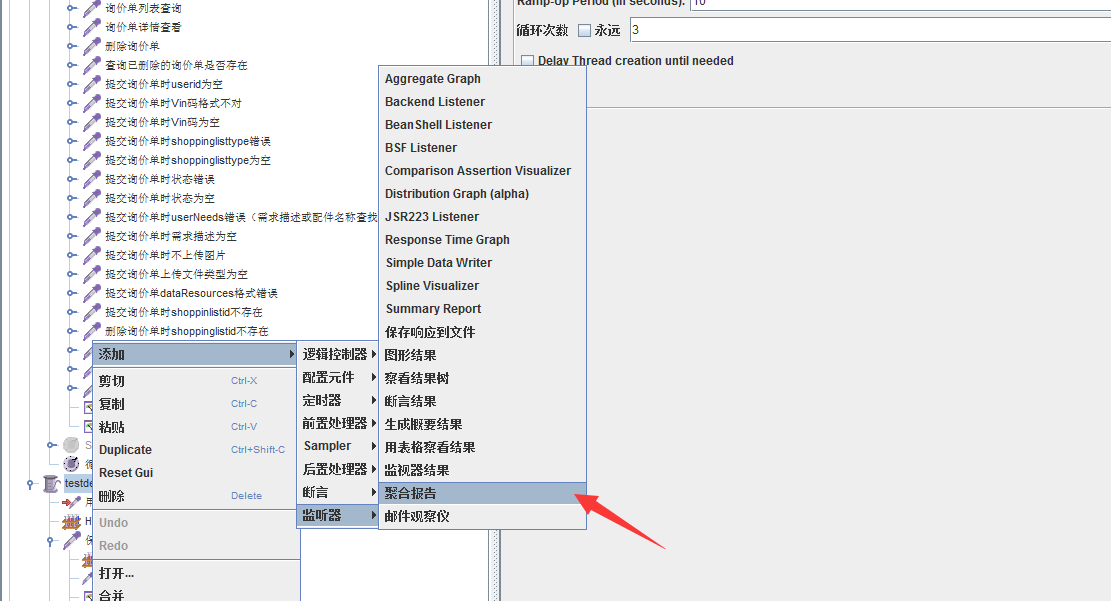
4、预期结果判断和执行结果查看&报告

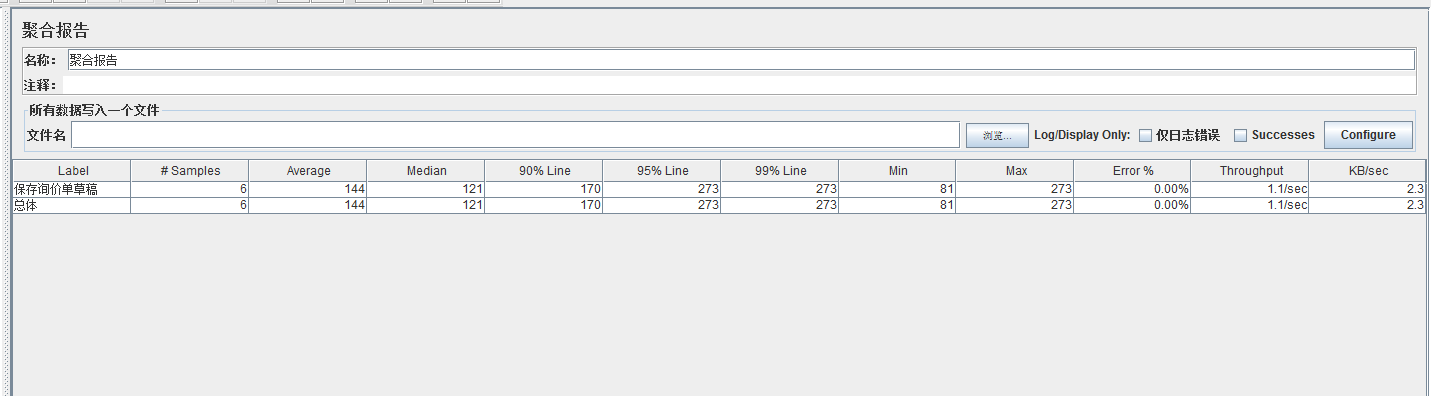
添加查看结果树，用于查看执行结果





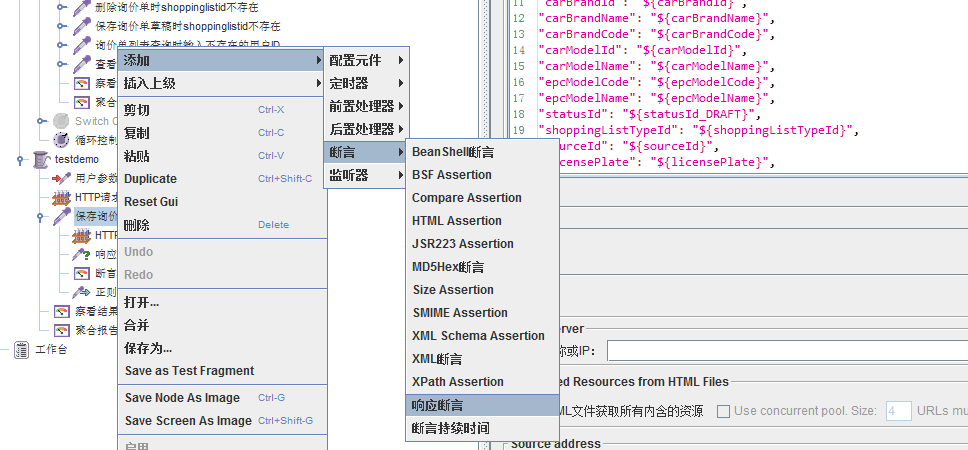
聚合报告添加，用于查看取样执行情况，主要是响应时间



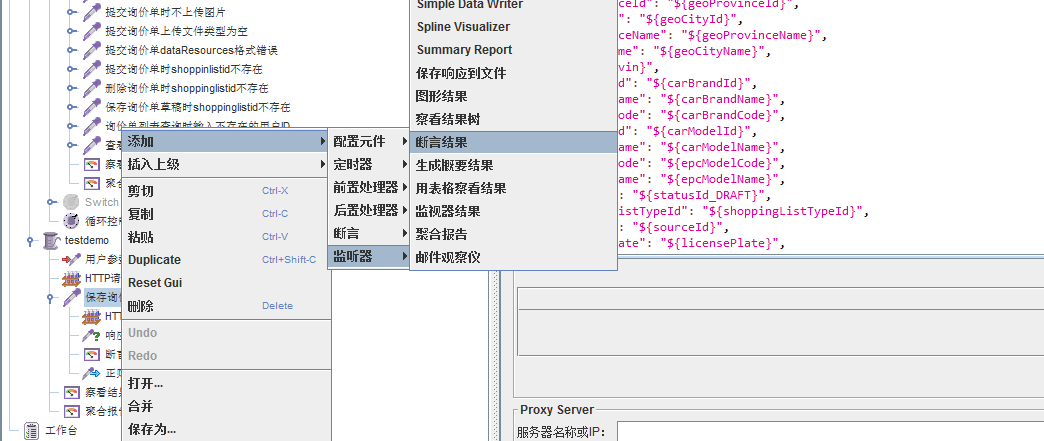


添加断言和断言响应

根据不同需求添加需要的断言



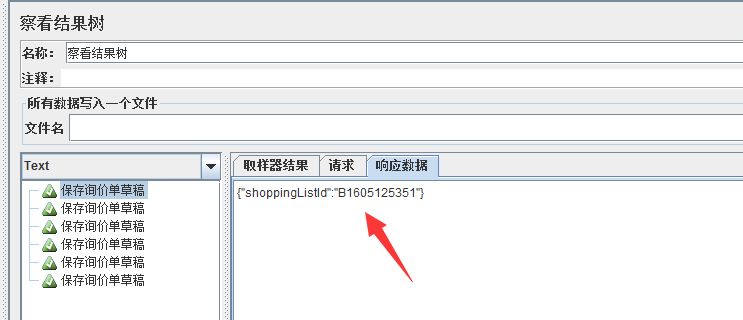
断言结果用于查看断言是否正确



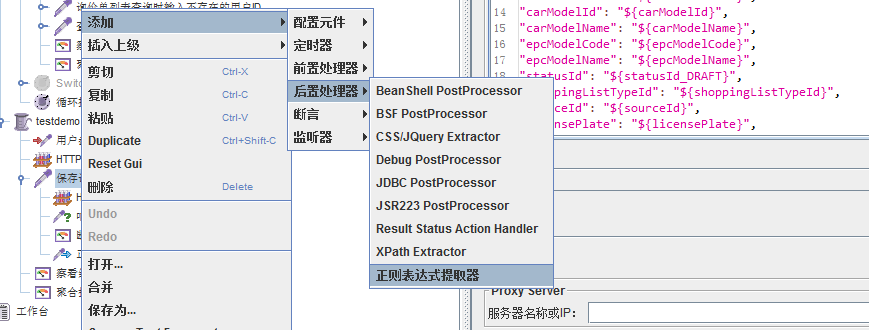
5、用例关联

如果有一些用例的输出是其下一个用例的输入的时候就应该做关联。

例如：在提交订单取样中我们提交成功了，返回了订单编号，在下一个取样中我们执行查询订单操作，恰好可以验证提交成功后返回的订单是否存在。



添加正则表达式提取器



这里有几个地方需要注意

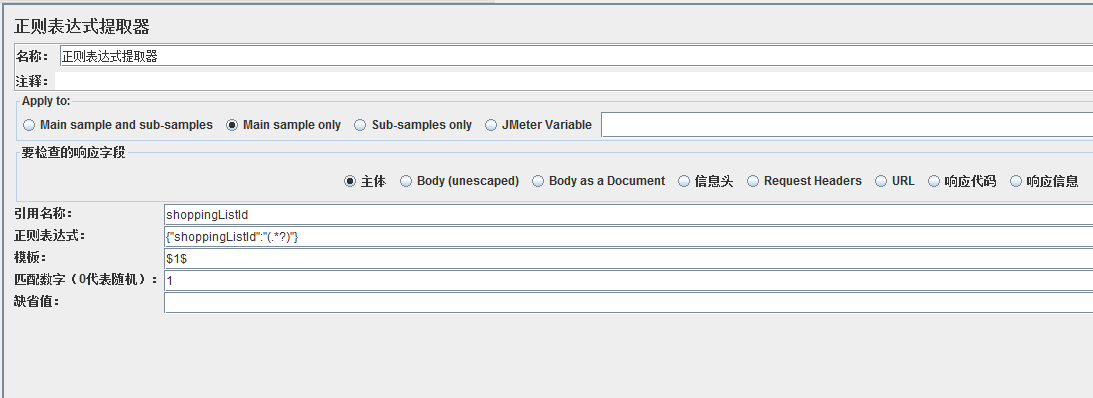
检查的响应字段，这里列了几个中，主体、body、消息头等等，根据需要选择你要取内容来源

引用名：定义的变量名称

正则表达式范例：{"shoppingListId":"(.\*?)"} 表示取到字符串{"shoppingListId":"和字符串"}之间的字符，中间部分()表示要提取的内容，.表示匹配任意字符串，\*表示任意长度，?表示找到第一个匹配项后终止，关于正则表达式就自己去理解吧。

模板：$1$ 表示取第一次匹配到的值

匹配数字：取到的结果内的值，0随机值 1全部值



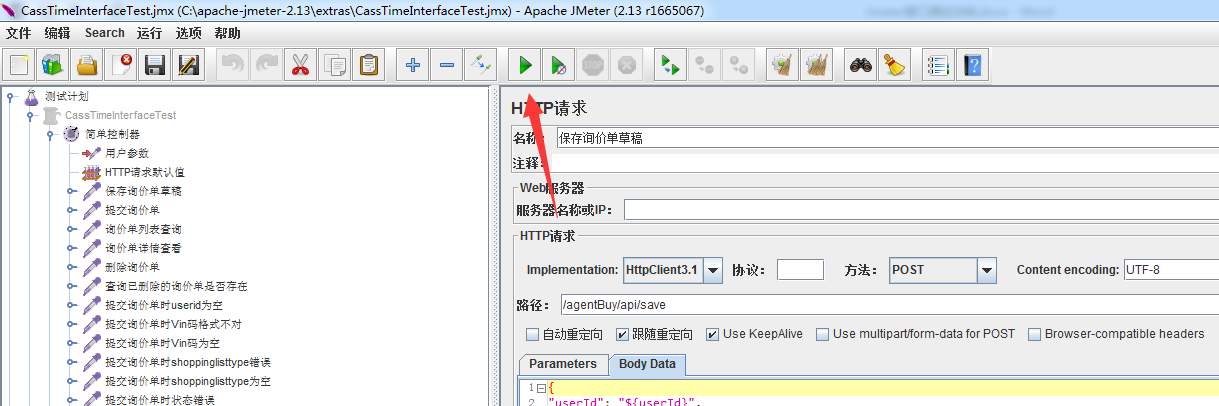
好了，一个基本流程的用例，就到此结束了，其中添加的一些元件中的没有介绍到的参数，还有未使用到的元件、控制器、监听器等等，在后续的实践中慢慢补全吧。

## jmeter soap取样范例

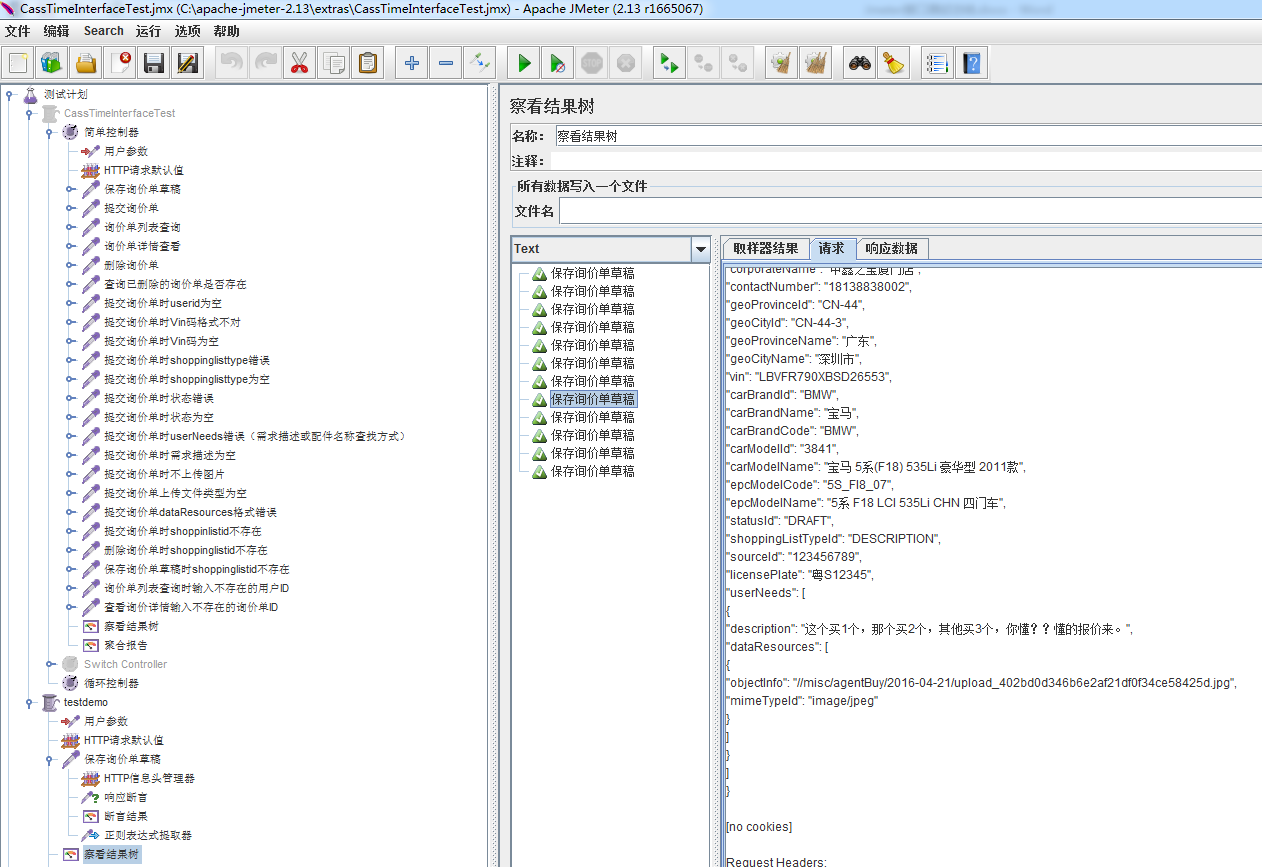
待实践

## 批量执行 jmeter 脚本并生成报表

用例已经完成了，可以开始执行用例，查看执行结果了，我们可以通过点击工具正上方的启动按钮来执行



执行之后的结果可以通过查看结果树和聚合报表来观察

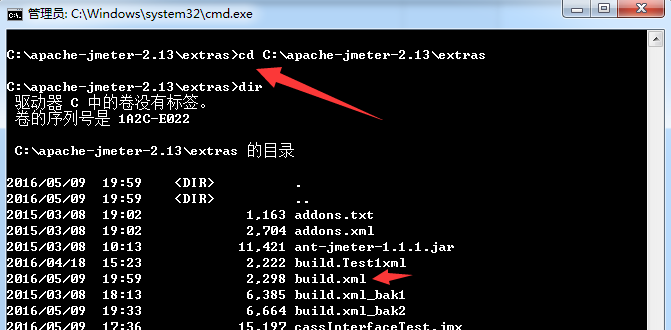


如果有很多用例需要执行，工具上的执行结果看起来就有点困难了，需要点来点去，有没有一个报表把执行情况和综合显示呢，答案是肯定的。

在此之前我们已经安装了ant，它是一个批量执行的工具，我们可以通过它来执行jmeter脚本，脱离图形化界面，当然这看起来好像并没有什么卵用，因为jmeter本身也能批量执行。有没有用先用了再说。

1、ant 批量执行命令

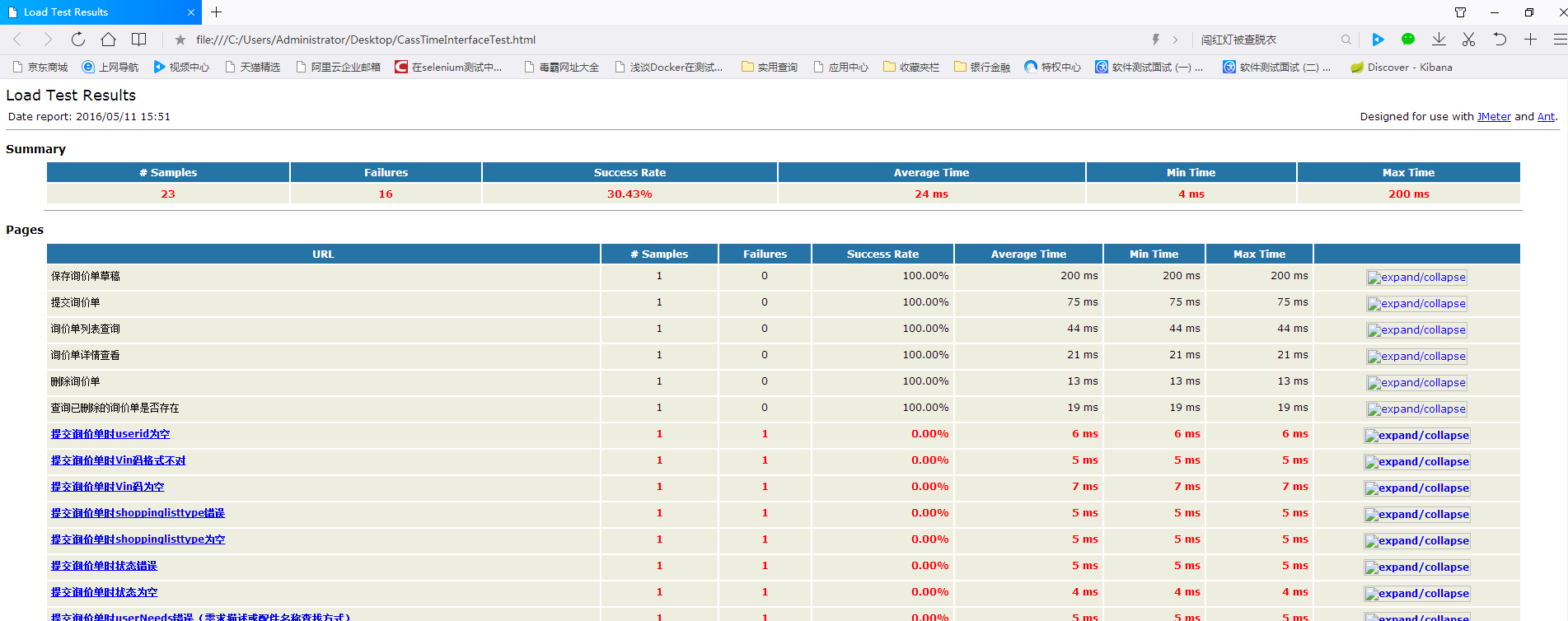
cmd打开命令运行窗口，跳转到jmeter 下build.xml文件存放路径



然后直接输入ant 执行build.xml 文件，成功执行后会有如下显示



执行完了，就看生成的报表了，生成HTML的报表我执行ant命令的目的之一。



默认的测试结果报表放在C:\apache-jmeter-2.13\extras 目录下，和工程文件是同一个位置，报表样式文件同样是放在这个目录下jmeter-results-detail-report\_21.xsl

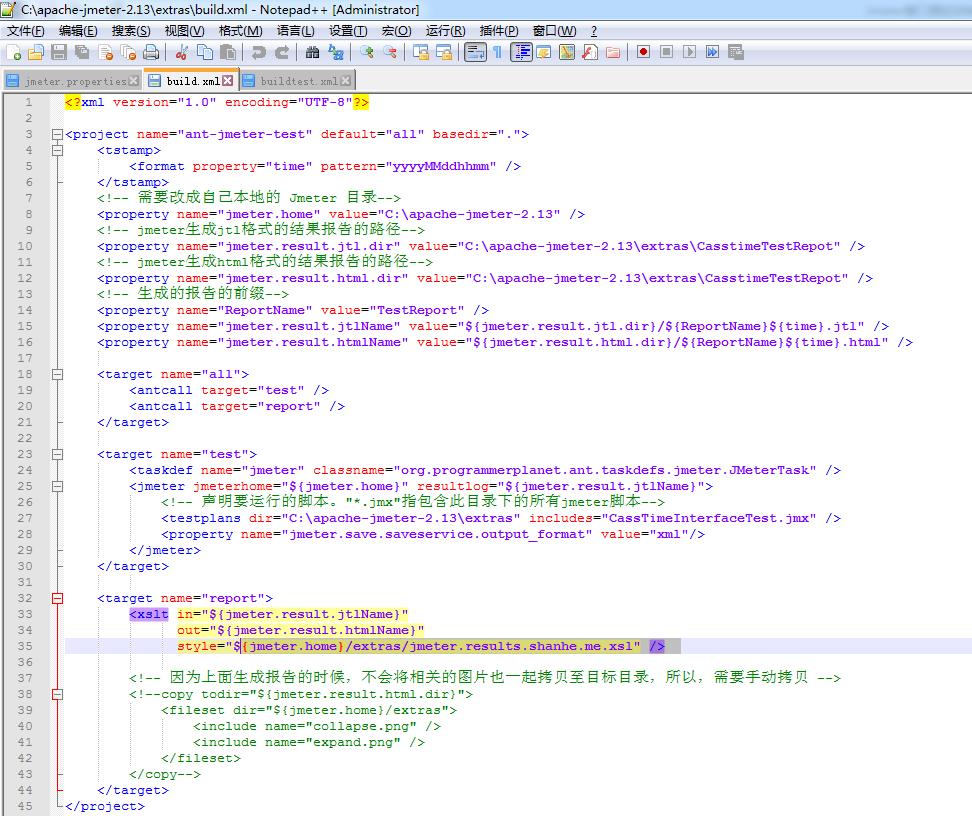
我们可以很清楚的看到报表中记录的信息是否使我们想要的。很明显，报表中的展示的内容不全，而且不易观察。出现的用例的问题也不容定位到详细错误，所以需要做一个报表优化。

## 报表优化

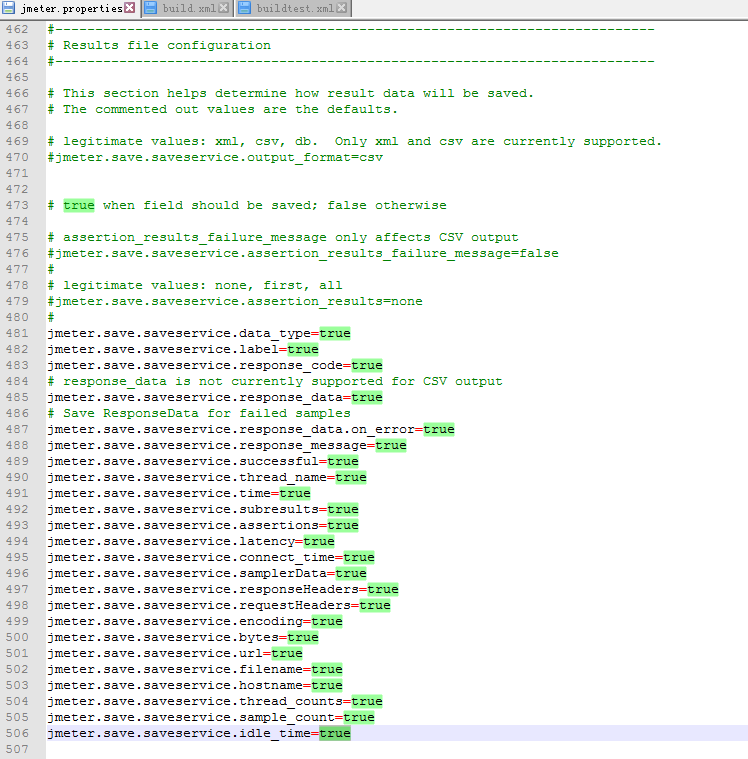
首先还是来看下上个章节中我们提到的build.xml，下面是我修改的build文件，根据需要自己去让ant帮你做事情。

build.xml路径C:\apache-jmeter-2.13\extras

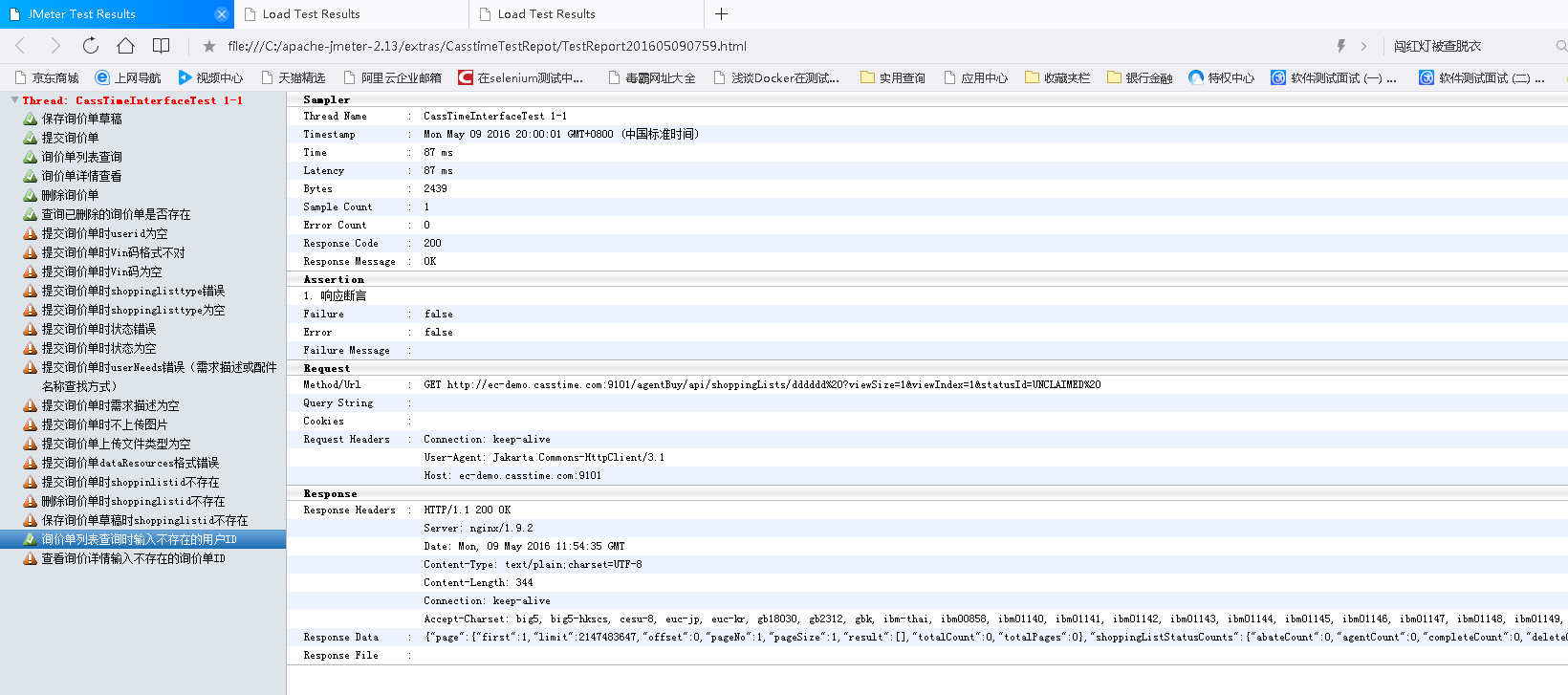
下面截图中声明了执行jmeter脚本生成的.jtl日志文件html报表生成的路径还有名称格式，要执行的脚本文件名称，还有报表的样式（这里提供一个比较综合性的报表\\192.168.29.218\Share\eCommerce\测试分\jmeter-results-detail-newreport.xsl，如果有兴趣也可以去修改或者寻找更合适的报表样式）



修改jmeter.properties文件中以下标注行的内容，这些是jmeter要输出的指标，用于记录在输出的报表上。



上面这些都修改完成之后再用ant去执行jmeter脚本就会生成下面这种报表了，这种方式相对来说就比较好看了，但是还有优化空间，后面的实践中会继续完善。



## Jmeter+Ant+Jenkins 持续集成接口测试平台

其实做完前面章节的所有内容还没达到最终目标，我们的目标是一键完成测试，生成可视化报告，而且所有人都能操作，都能查看。

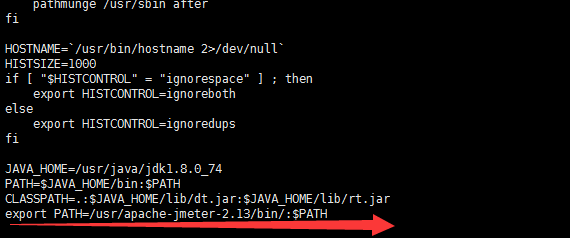
1. linux服务器上部署Jmeter

<http://jmeter.apache.org/download_jmeter.cgi> 上下载[apache-jmeter-2.13.tgz](http://apache.fayea.com/jmeter/binaries/apache-jmeter-2.13.tgz)

解压到/usr 下面，目录自己定

环境变量配置

在/etc 目录下修改profile文件，添加export PATH=/usr/apache-jmeter-2.13/bin/:$PATH

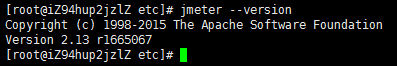


配置环境变量之后执行如下命令是环境变量生效。

#source /ect/profile

查看是否安装jmeter成功，检查jmeter版本

# jmeter –version



修改build.xml 和jmeter.properties，这个操作和Windows下面一样。

附送jmeter脚本执行命令一条

#jmeter -n -t baidu-test.jmx -l log.jtl 含义不赘述。

1. JDK安装

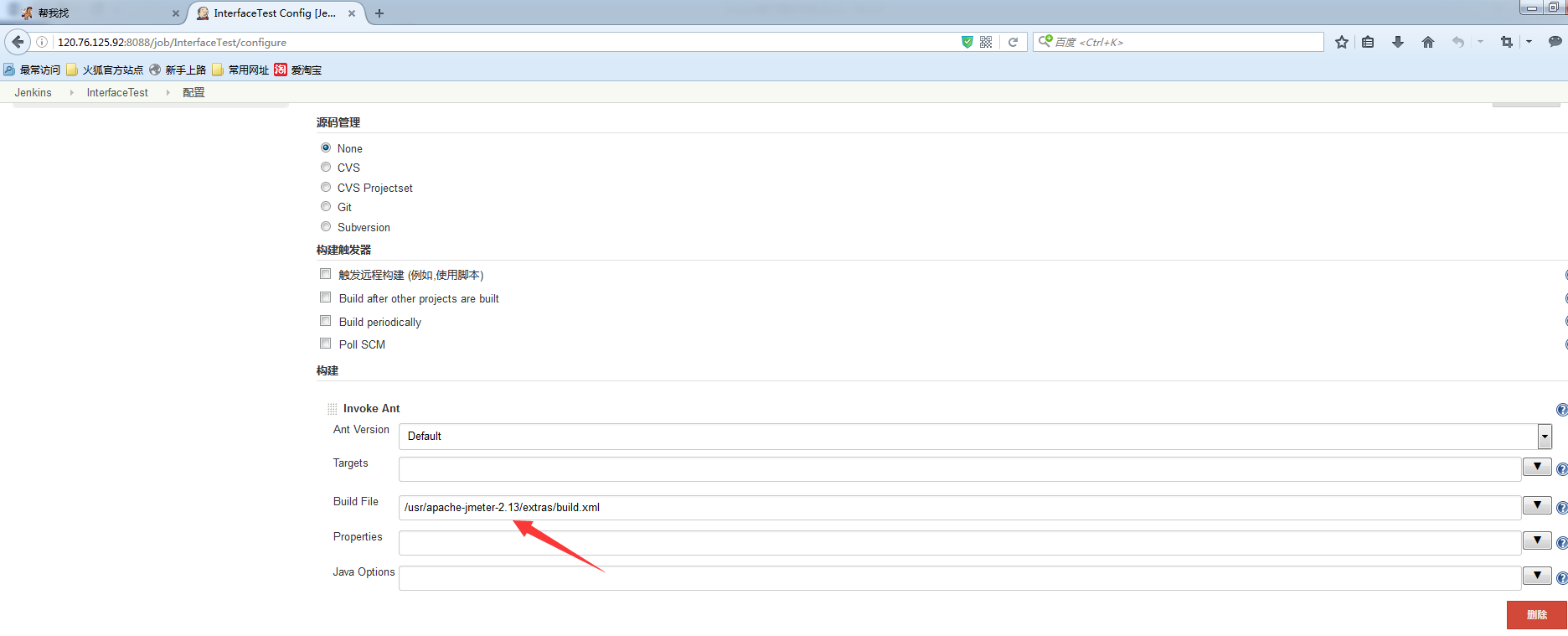
此处省略N个字。

1. Jenkins安装

新盼提供此部分内容，再次省略N个字。

1. 新建构建任务





除以上2个步骤外的其他设置，可根据需要去配置。然后就可以执行任务了。看起来好像挺简单的。其中有很多需要优化的地方，还有很多坑，会在后面的实践中继续完善。希望大家有新的知识点和实践内容都加到这个上面，让测试变的更专业，更轻松。

1. Jmeter持续集成Jenkins任务报表可视化处理

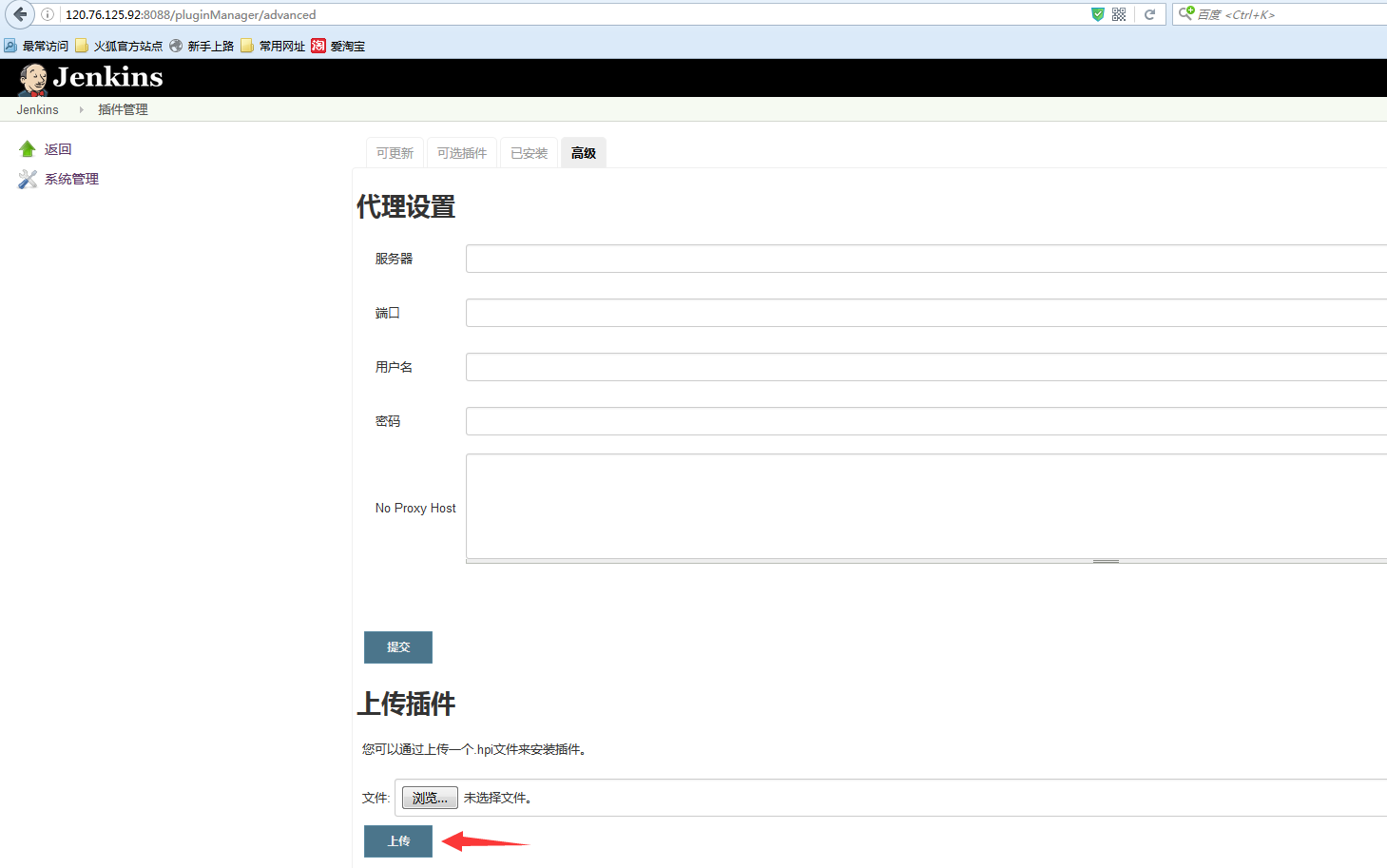
插件安装Performance Plugin、HTML Publisher Plugin

插件可以在系统管理里面直接安装也可以去官网下载，我是去官网下载的

https://jenkins.io/index.html

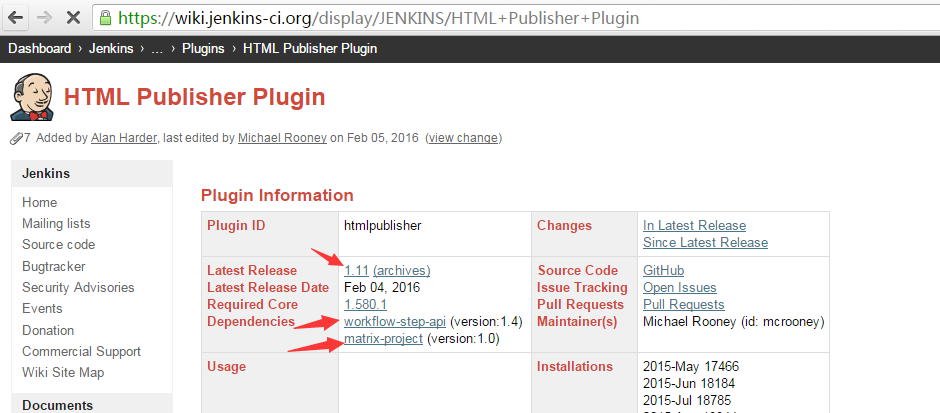
下载好之后在插件管理中选择高级直接上传插件文件就会自动安装了



在安装过程中可能会报一些xxx不存在的错误，如下图，是缺少依赖的插件，直接去下载对应插件安装。

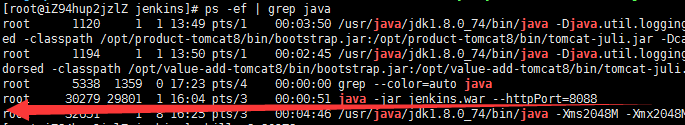


下载插件时，该插件说明中描述了依赖的插件



安装完插件之后需要重启Jenkins生效

ps -ef | grep java



kill -9 30279

Jenkins目录下执行启动命令

nohup java -jar jenkins.war --httpPort=8088 &

生效之后需要在构建任务中配置构建后操作



1. Jenkins 邮件发送处理

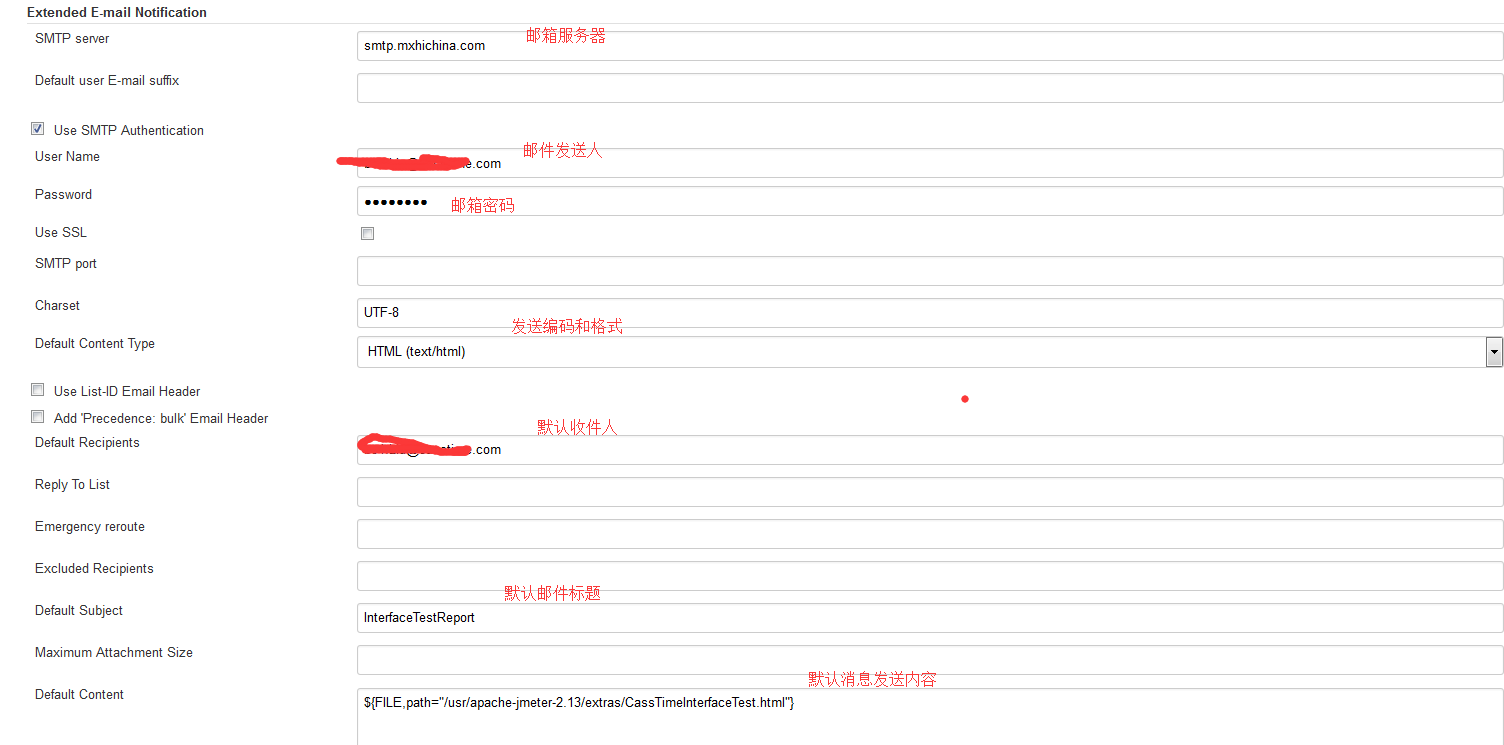
安装Jenkins邮件处理插件[Email Extension Plugin](https://wiki.jenkins-ci.org/display/JENKINS/Email-ext+plugin)，安装时也可能存在依赖插件需要安装，自己看着来。



安装插件后主页面选择 系统管理>系统设置

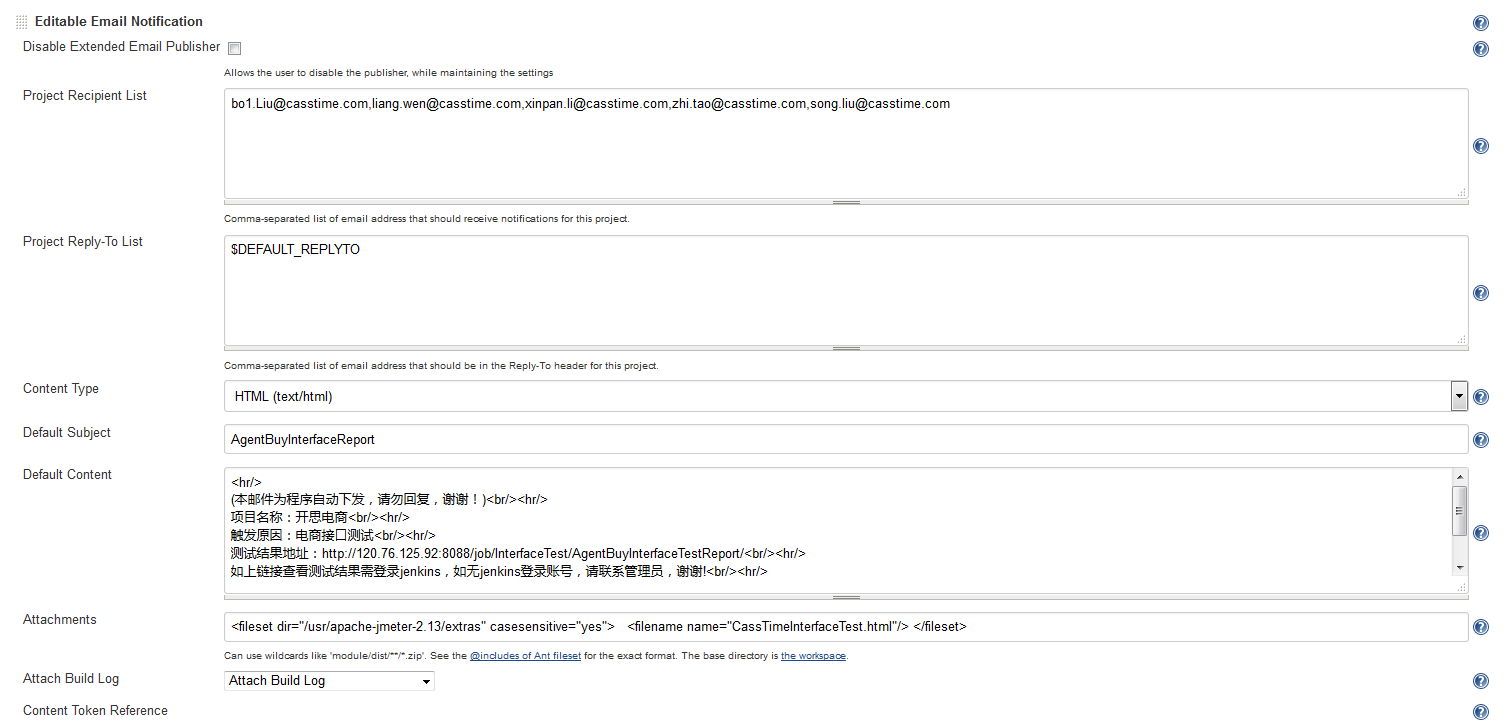


配置邮箱服务器



构建后步骤中添加默认配置

主要是配置收件人，其他配置在触发器中也会配置。





构建任务中配置触发器

1、触发规则（默认为总是触发） 2、发件人 3、邮件内容格式 4、邮件名称 5、邮件内容 6、邮件附件

