jmeter 入门系列

作者：苦叶子

Email: lymking@foxmail.com

公众号：DeepTest 或 开源优测

二维码:

关注公众号

1、获取最新版电子书，python selenium、jmeter 等系列

2、定期或不定期推送已完成写作的原创技术文章

公众号说明：

1、 目前已完成 python selenium 自动化测试系列

2、 Jmeter 综合实践系列正在进行中

3、 Python selenium 企业实践自动化测试框架设计系列正在规划中

声明：

第一篇 JMeter 之旅

1.1 前言

在你的日常工作中， 你有没有测试过一个服务能支撑多少用户在线，用户并发？

有没有那么一天，突然跟你说，晚上系统上线你做下性能测试。

1.png

1.2 什么是 JMeter

[本节内容来源百度百科^_^]

Apache JMeter 是 Apache 组织开发的基于 Java 的压力测试工具。用于对软件做压力测试，它最初被设计用于 Web 应用测试，但后来扩展到其他测试领域。 它可以用于测试静态和动态资源，例如静态文件、Java 小服务程序、CGI 脚本、Java 对象、数据库、

FTP 服务器， 等等。JMeter 可以用于对服务器、网络或对象模拟巨大的负载，来自不同压力类别下测试它们的强度和分析整体性能。另外，JMeter 能够对应用程序做功能/回归测试，通过创建带有断言的脚本来验证你的程序返回了你期望的结果。为了最大限度的灵活性，JMeter 允许使用正则表达式创建断言。

Apache jmeter 可以用于对静态的和动态的资源（文件，Servlet，Perl 脚本，java 对象，数据库和查询，FTP 服务器等等）的性能进行测试。它可以用于对服务器、网络或对象模拟繁重的负载来测试它们的强度或分析不同压力类型下的整体性能。你可以使用它做性能的图形分析或在大并发负载测试你的服务器/脚本/对象。

1.3 为什么是 JMeter

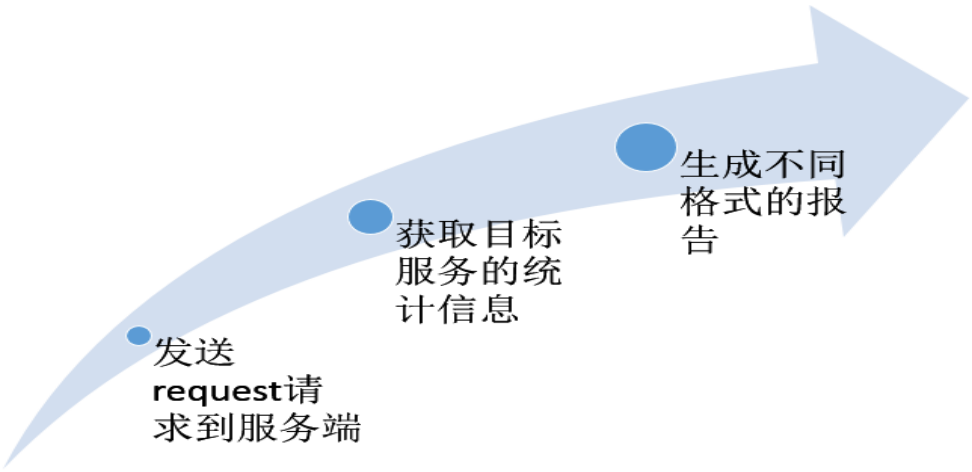
为什么选择 JMeter，下面看看 JMeter 的特色。

1. 开源许可: Jmeter 是完全免费的，并提供了源码可供自定义开发
2. 图形界面模式：提供了方便的图形界面来编辑和开发测试脚本
3. 平台无关：可以轻易在 windows、linux、mac 上运行
4. 多线程框架：通过线程组，能够轻易的设置不同测试的并发用户。

5. 图形测试结果：提供了图表、表格、树、文件等格式的结果显示。
6. 易于安装：jmeter 不需要安装，下载解压即可用。
7. 高扩展性：jmeter 支持用户自定义测试脚本，同样还提供了各种插件。
8. 多测试类型支持：支持性能测试、分布式测试、功能测试
9. 仿真模拟：支持多用户并发测试
10. 多协议支持：支持 http、jdbc、ldap、soap、jms、ftp 等等协议
11. 录制&回放：支持用 badboy 或 jmeter 录制，不过笔者从来不用该模式，纯手工最佳。
12. 脚本测试:jmeter 支持 beanshell 和 selenium

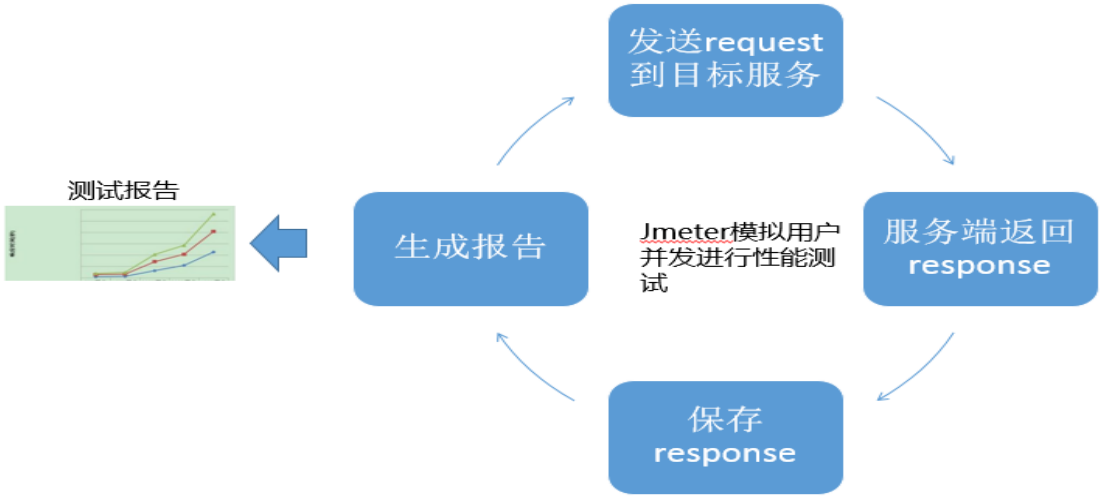
1.4 JMeter 工作原理

JMeter 基本工作原理如图：



2.png

JMeter 完整的工作原理如图：



3.png

1.5 总结

本次对 jmeter 进行了简单的基本介绍，主要让大家对 jmeter 有个基本的了解。

第二篇 JMeter 目录及关键配置分析 02

2.1 前言

学习一种工具，首先得对其关键配置及目录等有一个基本的了解，这样能更方便的深入掌握该工具，下面我们就 JMeter 的目录及相关关键配置进行分析说明。

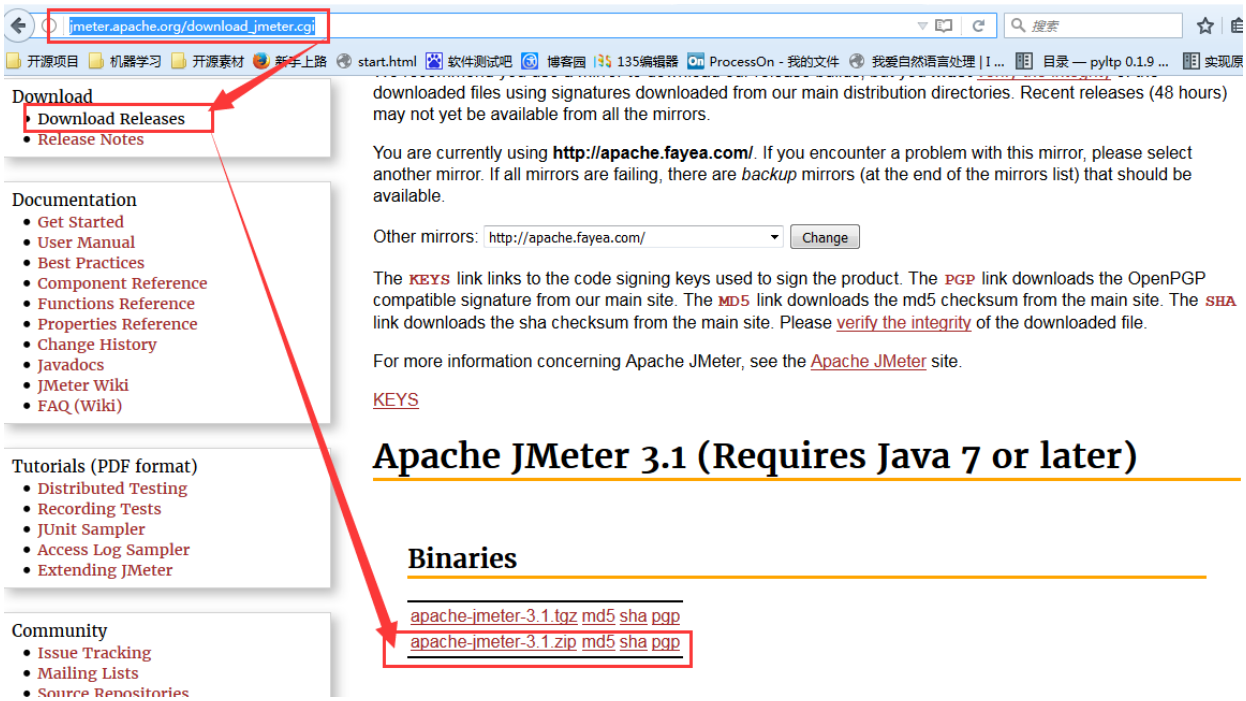
2.2 下载安装

1. 安装主程序

从 Apache JMeter 官网下最新版本：

http://jmeter.apache.org/download_jmeter.cgi

如图：



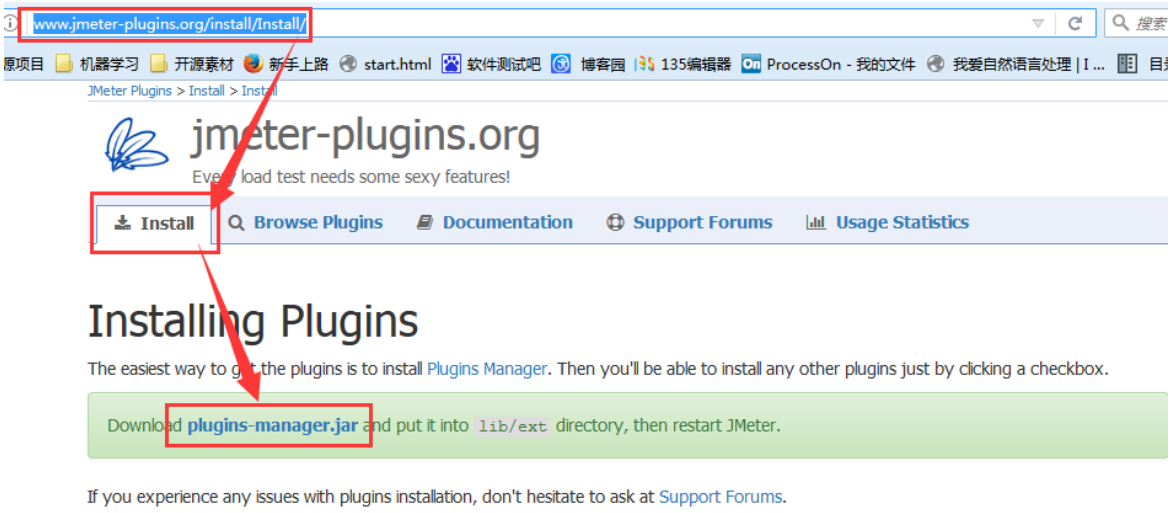
4.png

下载后直接解压即可。

1. 安装插件管理

从 http://www.jmeter-plugins.org/install/Install/ 下载插件管理包，如图：

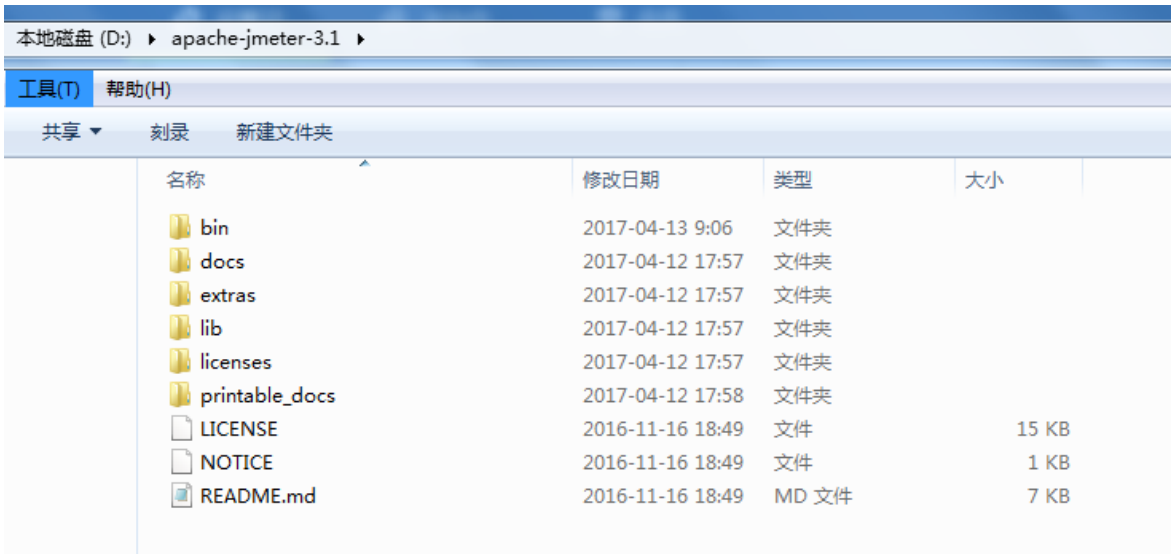
将下载的包放至 jmemter 解压根目录的 lib/ext 下即可。



6.png

2.3 目录说明

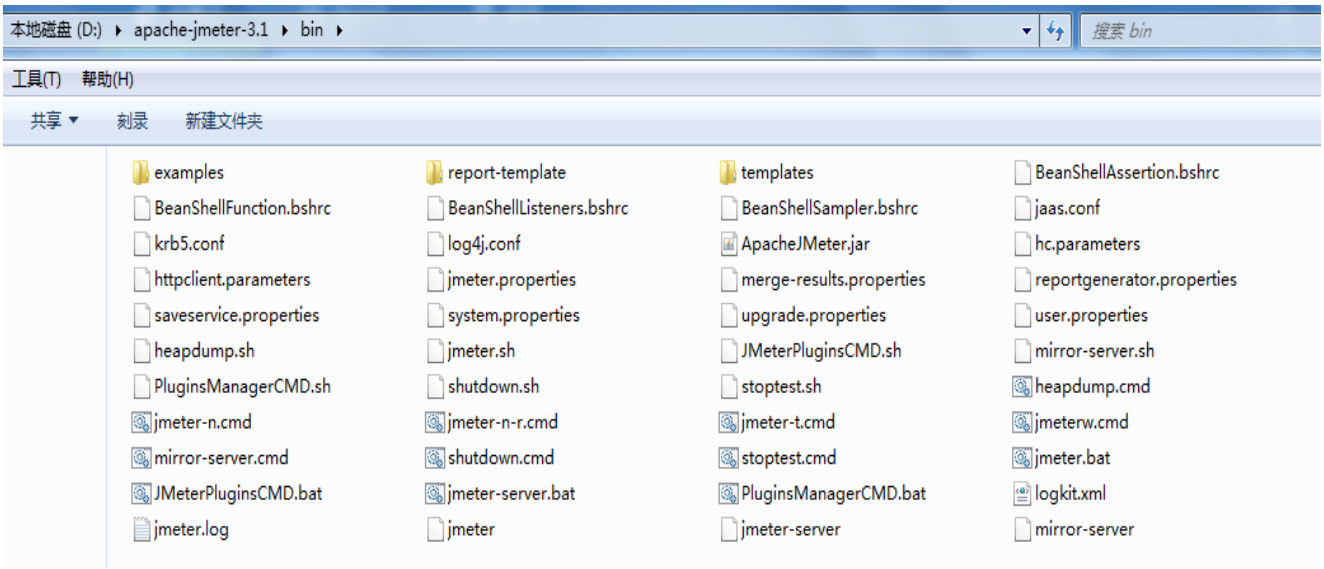
先看一下解压后的 JMeter 安装目录：



5.png

- 目录说明
- bin 包含启动、配置等相关命令
- docs 官方本地文档目录
- extras 辅助库
- lib 核心库，包含 JMeter 用到的各种基础库和插件
- licenses 包含 non-ASF 软件的许可证
- printable_docs 可打印版本文档目录
- LICENSE JMeter 许可说明
- NOTICE JMeter 简单信息说明
- README.md JMeter 官方基本介绍

下面我们重点看下 bin 目录，如图：



7.png

主要介绍 bin 目录下我们最关注几个文件：

- jmeter.properties JMeter 核心配置文件，各种配置基本在这完成
- log4j.conf JMeter 日志配置管理
- jmeter.log JMeter 运行日志记录，什么输出信息、警告、报错都在这里进行了记录
- jmeter.bat windows 下 jmeter 启动文件
- shutdown.cmd windows 下 jmeter 关闭文件
- stoptest.cmd windows 下 jmeter 测试停止文件
- jmeter-server.bat windows 下 jmeter 服务器模式启动文件

==注：每一个.cmd 文件都对应一个.sh 文件，.sh 是 linux 下的对应功能的文件==

其他文件的功能就不一一说明了，同时其他目录这里也不再进行阐述，有兴趣的朋友可以自己深入看下。

2.4 关键配置说明

1. jmeter.properties 配置说明

主要包含以下几个方面的配置：

- SSL 配置：

重点关注下面几个配置

```
# 指定 HTTPS 协议层
https.default.protocol=TLS

# 指定 SSL 版本，实际应用中可能需要修改
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
```

- JMeter 界面显示配置

这里就不对其界面显示控制进行说明了，一般情况下默认界面能满足大家的应用了。

- JMeter 测试项目自动备份配置

```
# 设置是否启用自动备份，默认是 true
jmeter.gui.action.save.backup_on_save=true

# 设置自动备份目录，默认备份至 JMeter 根目录的 backups 下
jmeter.gui.action.save.backup_directory=

# 设置自动备份项目数，默认为最近 10 个
jmeter.gui.action.save.keep_backup_max_count=10
```

- 远程主机配置

```
# 配置远程主机的 IP，默认为本机。用逗号","可以设置多个远程主机
remote_hosts=127.0.0.1
# 多个远程主机指定示例如下,其中:后为端口
remote_hosts=127.0.0.1:1099,127.0.0.1:1200,127.0.0.1:1300
```

对于 RMID 的配置请直接看配置文件中的选项说明

- 日志管理配置

设置日志格式

log_format_type=default

设置日志输出级别

log_level.jmeter=INFO

设置 junit 日志输出级别

log_level.jmeter.junit=DEBUG

设置日志输出目标文件，默认为 jmeter.log

log_file=jmeter.log

- 等等其他还有 10 多个配置大项（就不一一列举了）
- jmeter.bat 关键配置修改

为了更优化的使用 jmeter，需要对 jmeter.bat 中的一些配置根据当前机器的配置进行优化，这里进行关键配置项说明，大家根据自己的机器的配置来进行修改。

jvm 相关配置，大概在 80 行左右，找到这些配置，

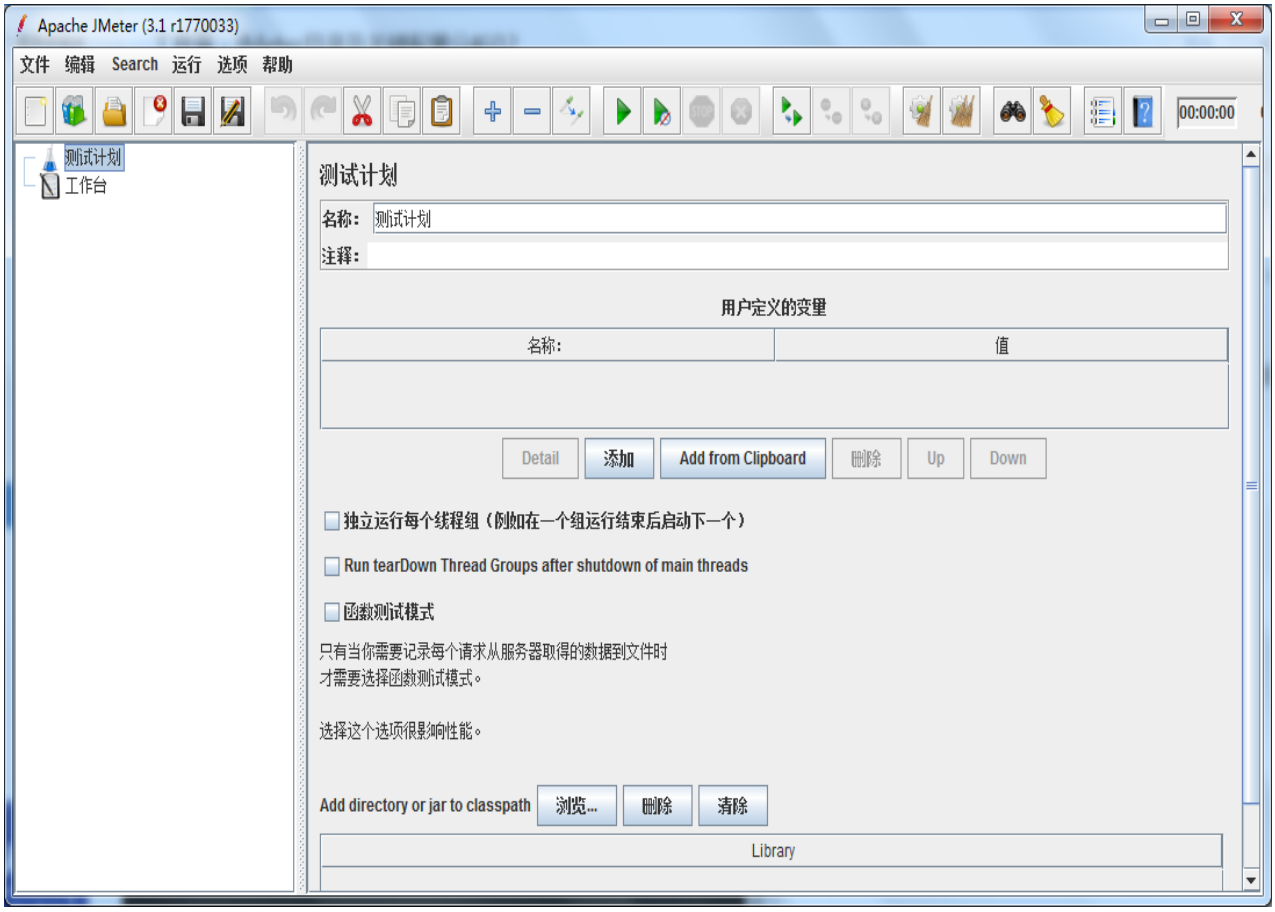
对其中的数值根据当前机器的硬件配置来修改。

```
    set HEAP=-Xms2048m -Xmx2048m
    set NEW=-XX:NewSize=512m -XX:MaxNewSize=512m
    set SURVIVOR=-XX:SurvivorRatio=8 -
XX:TargetSurvivorRatio=50%
    set TENURING=-XX:MaxTenuringThreshold=2
if %current_minor% LEQ "8" (
     rem Increase MaxPermSize if you use a lot of Javascript in
your Test Plan :
     set PERM=-XX:PermSize=512m -
XX:MaxPermSize=1024m
)
```

2.5 启动 jmeter

在 bin 目录下直接双击 jmeter.bat 即可

启动后的界面如下：



8.png

2.6 总结

本次就 jmeter 的安装和配置及关键配置项进行了分享，大家可以深入的去研究下其他的一些配置，以便进一步的熟悉 jmeter 的原理和应用。

第三篇 HTTP 协议报文结构及示例

3.1 前言

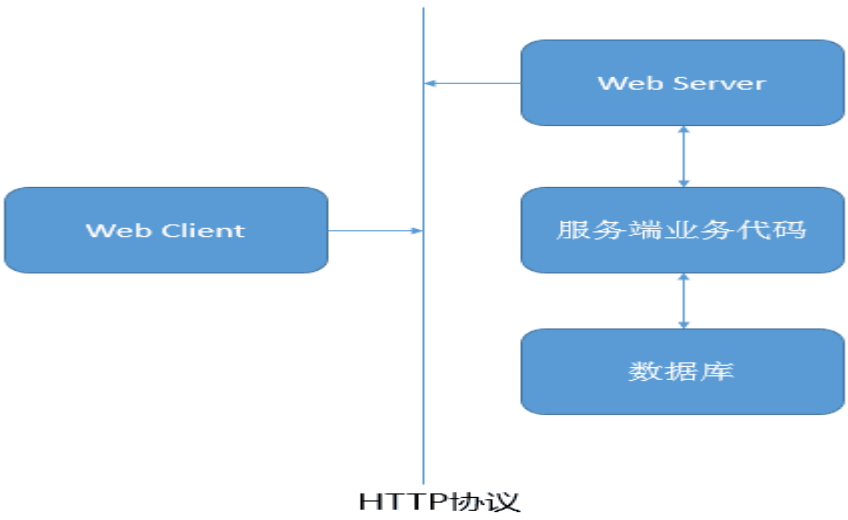
从事性能测试必不可绕过的就是协议，对基本知识的了解也还，还是深入掌握协议的机制，都能让你在从事性能测试实施时显得更加顺手。

下面我们就 HTTP 协议及性能测试过程必须掌握的一些分析工具来进行分享。

重点分享性能测试实施过程中必须掌握的关键技术、工具。更细节的请参考 HTTP 相关书籍或 RFC 文档。

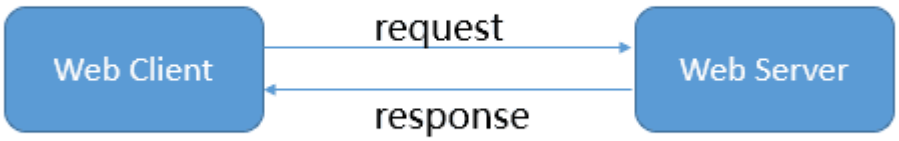
3.2 HTTP 基本架构

下面我们用一张简单的流程图来展示 HTTP 协议基本架构，以便大家先有个基本的了解。



- Web Client 可以是浏览器、搜索引擎、机器人等等一切基于 HTTP 协议发起 http 请求的工具。
- Web Server 可以是任何的能解析 HTTP 请求，并返回给 Web Client 可识别的响应的服务，常见的有 apache、nginx、IIS 等等 web 服务器。

浓缩就是精华，看下最简洁的 HTTP 交互图：



10.png

3.3 HTTP 报文结构

- 请求报文

HTTP 请求报文由请求行、请求头、空行和请求内容 4 个部分构成。

如下图所示：

11.png

下面对上图进行简单的分析：

请求行

由请求方法字段、URL 字段、协议版本字段三部分构成，它们之间由空格隔开。常用的请求方法有：GET、POST、HEAD、PUT、DELETE、OPTIONS、TRACE、CONNECT。

请求头

请求头由 key/value 对组成，每行为一对，key 和 value 之间通过冒号(:)分割。请求头的作用主要用于通知服务端有关于客户端的请求信息。

典型的请求头有：

User-Agent：生成请求的浏览器类型

Accept：客户端可识别的响应内容类型列表；星号* 用于按范围将类型分组。*/*表示可接受全部类型，type/*表示可接受 type 类型的所有子类型。

Accept-Language: 客户端可接受的自然语言

Accept-Encoding: 客户端可接受的编码压缩格式

Accept-Charset： 可接受的字符集

Host: 请求的主机名，允许多个域名绑定同一 IP 地址

connection：连接方式（close 或 keeplive）

Cookie: 存储在客户端的扩展字段

空行

最后一个请求头之后就是空行，用于告诉服务端以下内容不再是请求头的内容了。

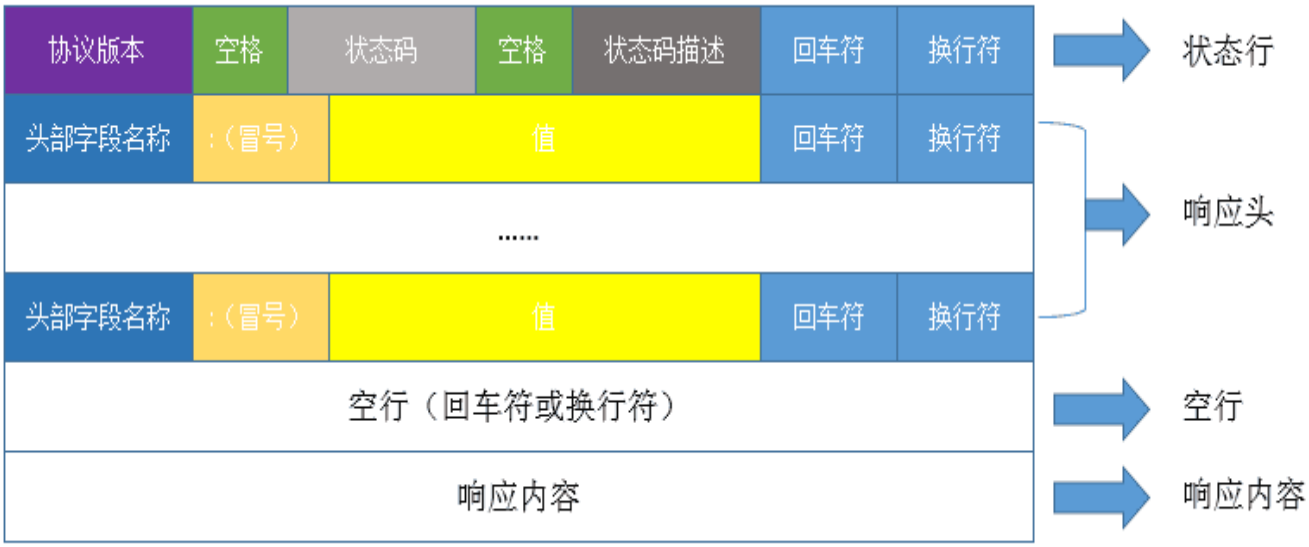
请求内容

请求内容主要用于 POST 请求，与 POST 请求方法配套的请求头一般有 Content-Type（标识请求内容的类型）和 Content-Length（标识请求内容的长度）

- 响应报文

HTTP 响应报文由状态行、响应头、空行和响应内容 4 个部分构成。

如下图所示：



12.png

下面对响应报文格式进行简要的分析说明：

状态行

由 HTTP 协议版本、状态码、状态码描述三部分构成，它们之间由空格隔开。

状态码由 3 位数字组成，第一位标识响应的类型，常用的 5 大类状态码如下：

1xx：表示服务器已接收了客户端的请求，客户端可以继续发送请求

2xx：表示服务器已成功接收到请求并进行处理

3xx：表示服务器要求客户端重定向

4xx：表示客户端的请求有==非法内容==

5xx：标识服务器未能正常处理客户端的请求而出现意外错误

常见状态码说明：

200 OK： 表示客户端请求成功

400 Bad Request： 表示客户端请求有语法错误，不能被服务器端解析

401 Unauthonzed： 表示请求未经授权，该状态码必须与 WWW-Authenticate 报文头一起使用

404 Not Found：请求的资源不存在，例如输入了错误的 url

500 Internal Server Error： 表示服务器发生了不可预期的错误，导致无法完成客户端的请求

503 Service Unavailable：表示服务器当前不能处理客户端的请求，在一段时间后服务器可能恢复正常

响应头

一般情况下，响应头会包含以下，甚至更多的信息。

Location：服务器返回给客户端，用于重定向到新的位置

Server： 包含服务器用来处理请求的软件信息及版本信息

Vary：标识不可缓存的请求头列表

Connection: 连接方式。

对于==请求端==来讲：close 是告诉服务端，断开连接，不用等待后续的求请了。keeplive 则是告诉服务端，在完成本次请求的响应后，保持连接，等待本次连接后的后续请求。

对于==响应端==来讲：close 表示连接已经关闭。keeplive 则表示连接保持中，可以继续处理后续请求。Keep-Alive 表示如果请求端保持连接，则该请求头部信息表明期望服务端保持连接多长时间（秒），例如 300 秒，应该这样写 Keep-Alive: 300

空行

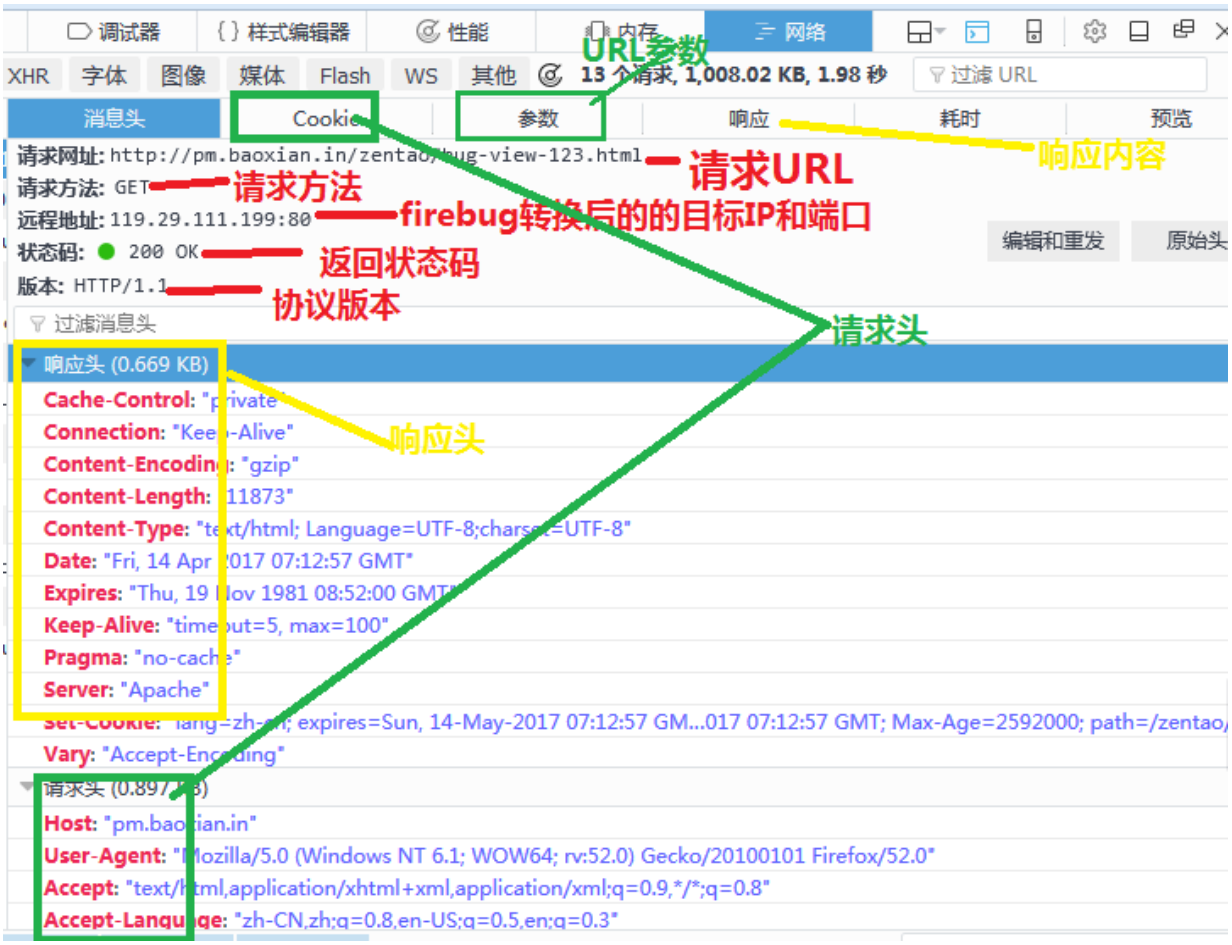
最后一个响应头之后就是空行，用于告诉请求端以下内容不再是响应头的内容了。

响应内容

服务端返回给请求端的文本信息。

3.4 HTTP 报文示例

在这里我们在 Firefox 下用 firebug 随意抓取一个 HTTP 包和上文的报文结构做下一一对应关系图，以便大家了解实际的包和标准报文结构的对应关系。



13.png

3.5 总结

对于 HTTP 协议的交互过程这里就不再进行说明了，大家可以搜索下相关的资料进行学习，上述的内容请务必熟练掌握、深刻了解。

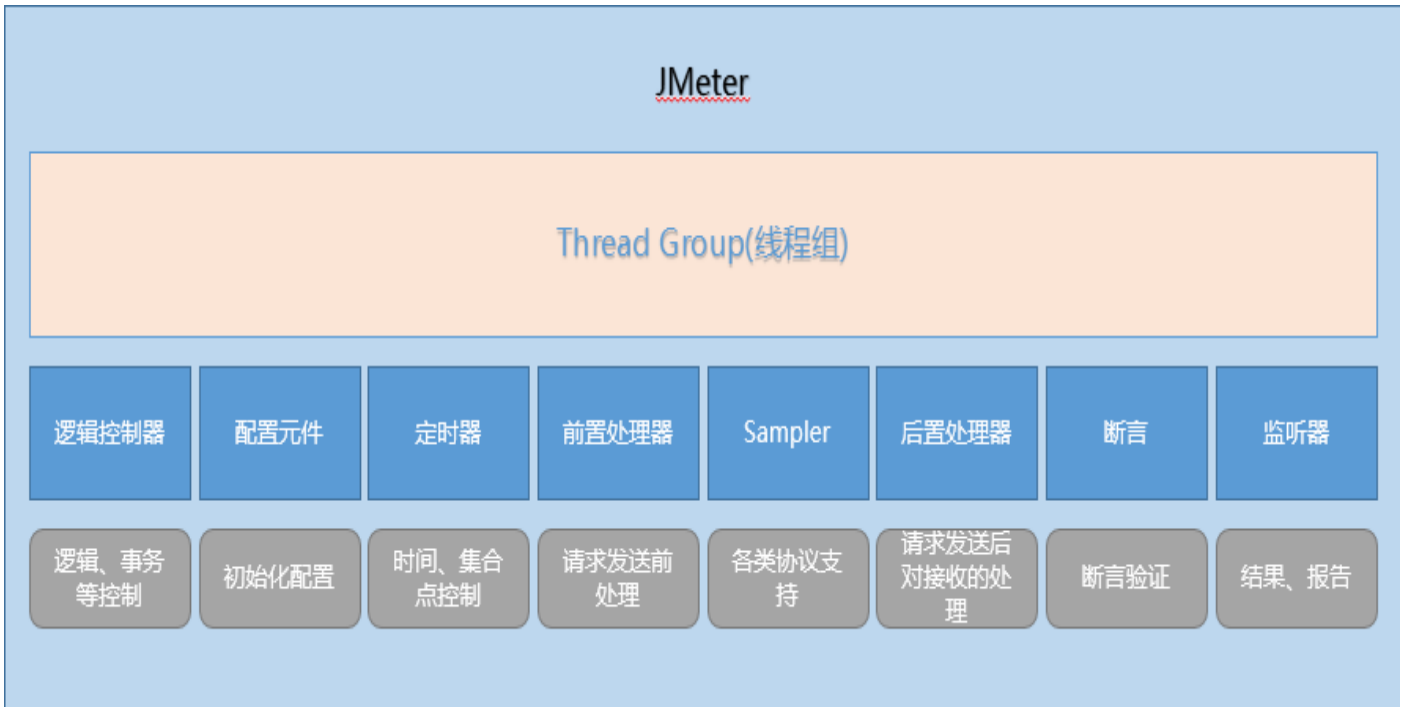
更详细的内容推荐大家学习 RFC 2616（http 协议 1.1 版本，有中文版本）

第四篇 JMeter 组件手册

4.1 前言

在 jmeter 中提供了一系列的不同的组件，每一种组件都提供了某类功能的实现，用于支持性能测试的实施。

请看下图，jmeter 的核心组件构成。



41.png

学习、研究 jmeter 之前，深入了解 jmeter 的基本组件及其作用是必须的。接下来我们开始讨论基于 jmetere 进行性能测试必须掌握的组件，以便大家逐步掌握 jemter 的核心基本能力。

下面的几个组件是入门 jmeter 必须掌握的：

- Thread Group
- Samplers
- Listeners
- Configuration

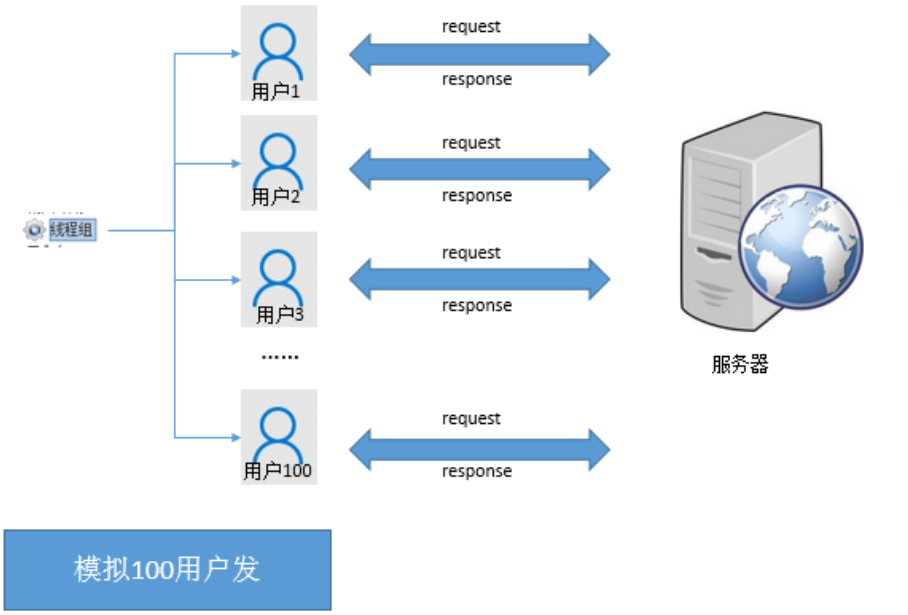
4.2 Thread Group(线程组)

线程组是一系列线程的集合，每一个线程代表着一个正在使用应用程序的用户。在 jmeter 中，每个线程意味着模拟一个真实用户向服务器发起请求。

在 jmeter 中，线程组组件运行用户设置线程数量、初始化方式等等配置。

例如，如果你设置线程数为 100，那么 jmeter 将创建并模拟测试 100 个用户请求到服务器端。

如下图所示：



42.png

4.3 Samplers

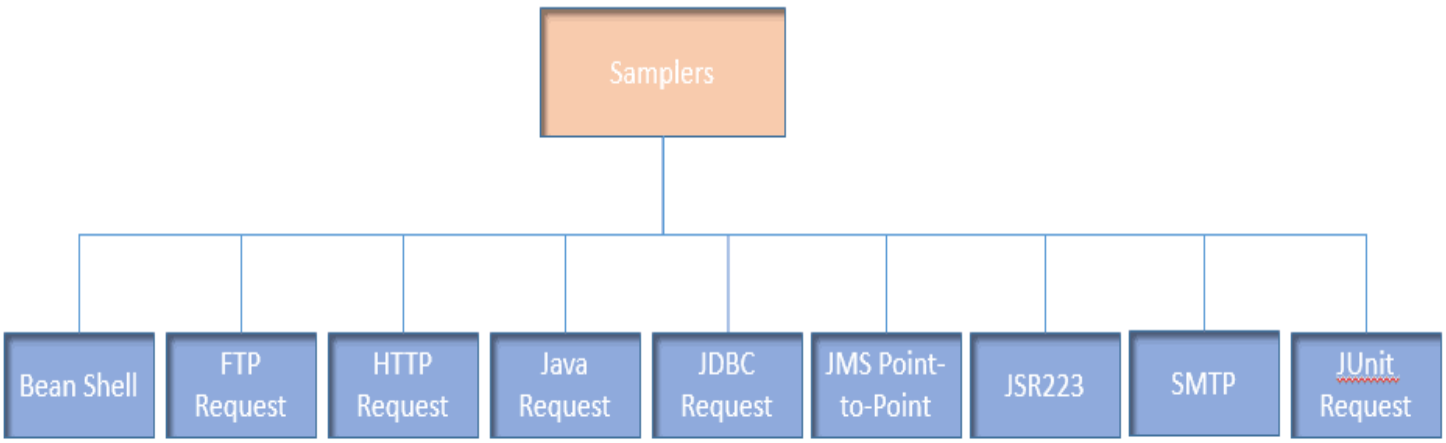
我们常用的 jmeter 测试有 HTTP、FTP、JDBC 协议，以及其他各种支持的协议。

在上节我们已经知道线程组件用于模拟用户请求至服务器端。

但还未讲解如何在线程组件中实现某种请求类型（比如如何发起 HTTP 请求？）。

在本节中，我们将演示如何利用 Samplers 组件的元素来实现各类请求类型。

我们先看一下在 jmeter 中 Samplers 组件已经实现了哪些协议的支持。如下图所示:



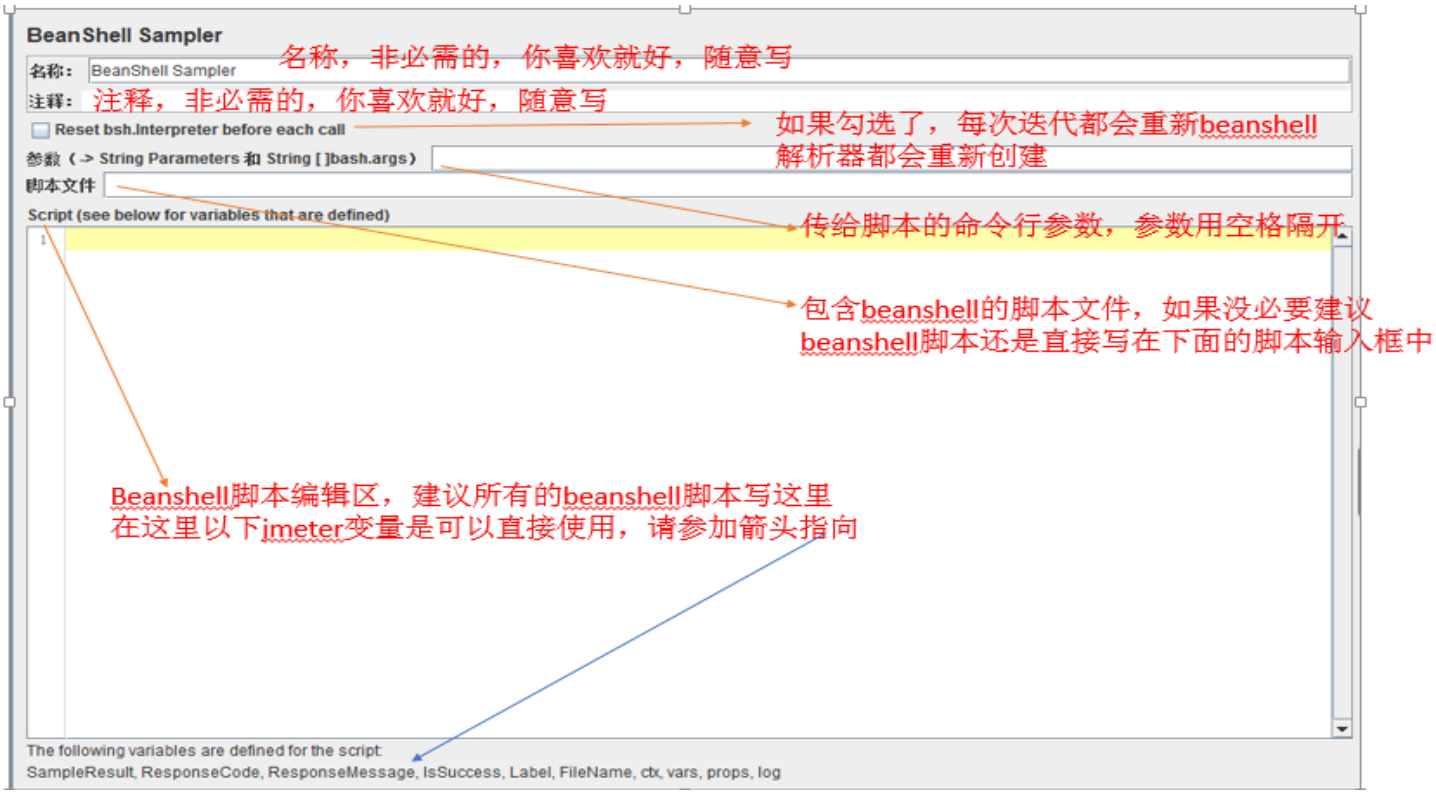
43.png

下面我们就重要的 Samplers 组件元素进行一一讲解，以便大家有个初步的了解。

- BeanShell Sampler

这个组件元素允许我们在 jmeter 中写 Bean Shell 脚本，写这个脚本有什么作用？意味着你可以完全的控制和实现自己的需要。灵活定制，自然也就有难度，你得有点脚本功底。

参见图说明：



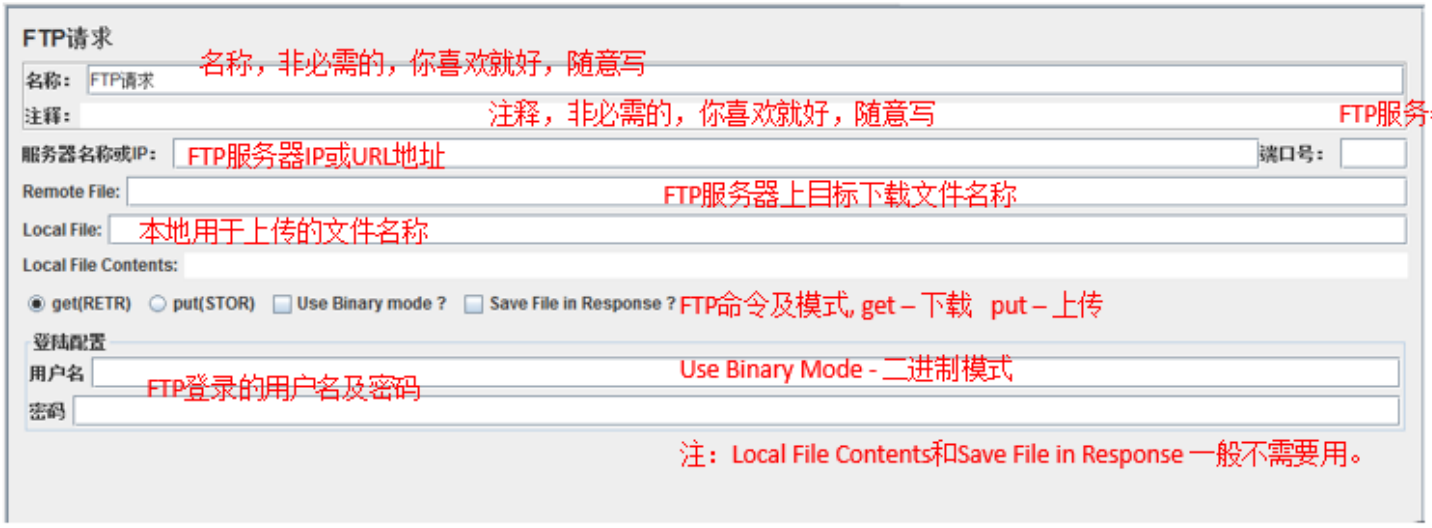
44.png

注：每一个 Sampler 都有自己独立的 beanshell 解析器，并且 sampler 只能在自己的线程中调用（意味着不可跨线程使用）。

- FTP Request

FTP Request 元素提供了测试 ftp 服务器的能力，这个元素让我们能够去测试 ftp 的上传、下载功能。

下面我们看一下 ftp 元素的基本配置说明：



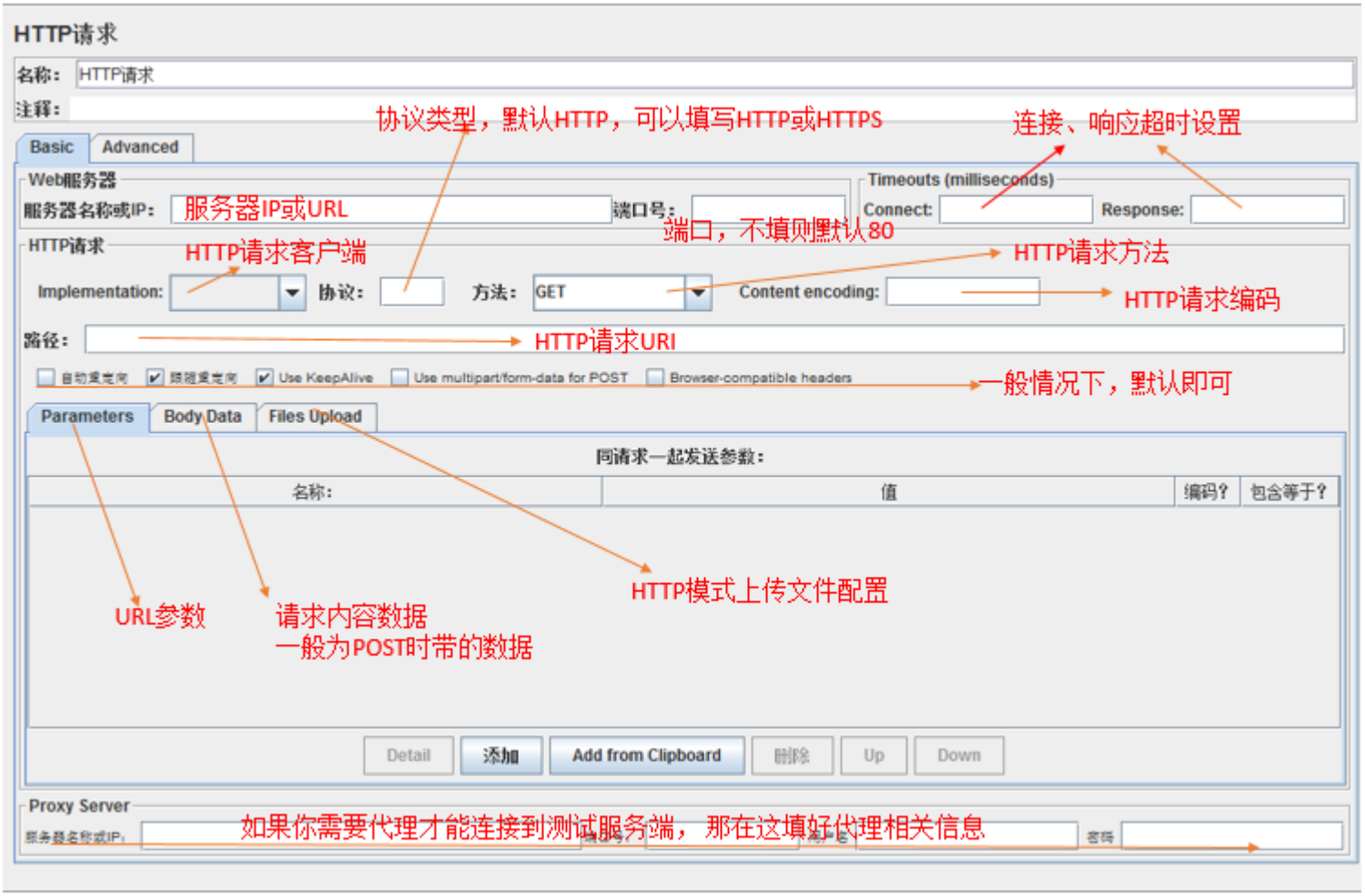
45.png

注：我们经常在 windows 和 linux 直接通过 ftp 进行文件传输，建议勾选 Use Binary Mode，避免编码问题。

- HTTP Request

HTTP Request 提供了 HTTP/HTTPS 协议的测试支持能力。

下面我们一起看看 HTTP Request 元素的基本配置说明，了解下基本的功能。



46.png

- Java Request

Java Request 提供了测试 java API 的支持，但要注意要测试的 java API 需要有对应的测试类，该测试类必须继承 AbstractJavaSamplerClient。

示例如下：

待测类 class Sum; -> 生成 sum.jar

继承至 AbstractJavaSamplerClient 的测试类 Class TestSum(AbstractJavaSamplerClient) -> 生成 testSum.jar

==注：==

一个 java 测试应该要实现以下几个方法，以便 jmeter java sampler 可以正确调用：

方法	说明
Arguments getDefaultParameters()	用于获取jmeter java sampler传入的测试数据
SampleResult runTest(JavaSamplerContext arg0)	测试事务处理
void setupTest(JavaSamplerContext arg0)	初始化
void teardownTest(JavaSamplerContext arg0)	清理

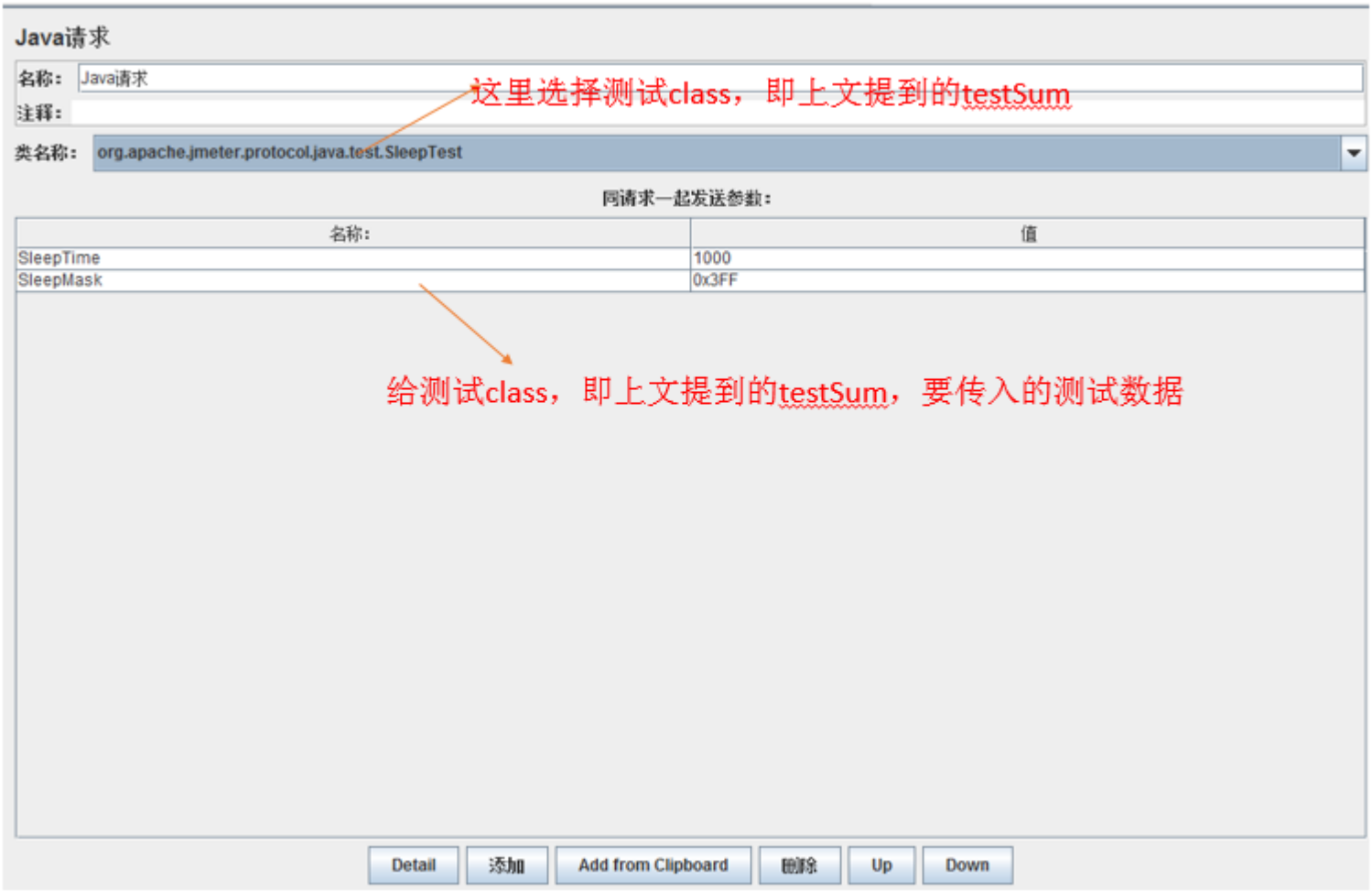
416.png

更详细的后续出专题讲解，本篇不举具体示例了。

注意 testSum.jar 要能调用 sum.jar。

将上述 sum.jar、testSum.jar 拷贝至 jmeter 安装目录的 lib/ext 下。

下面我们看看如何在 jmeter 配置 java 测试。



47.png

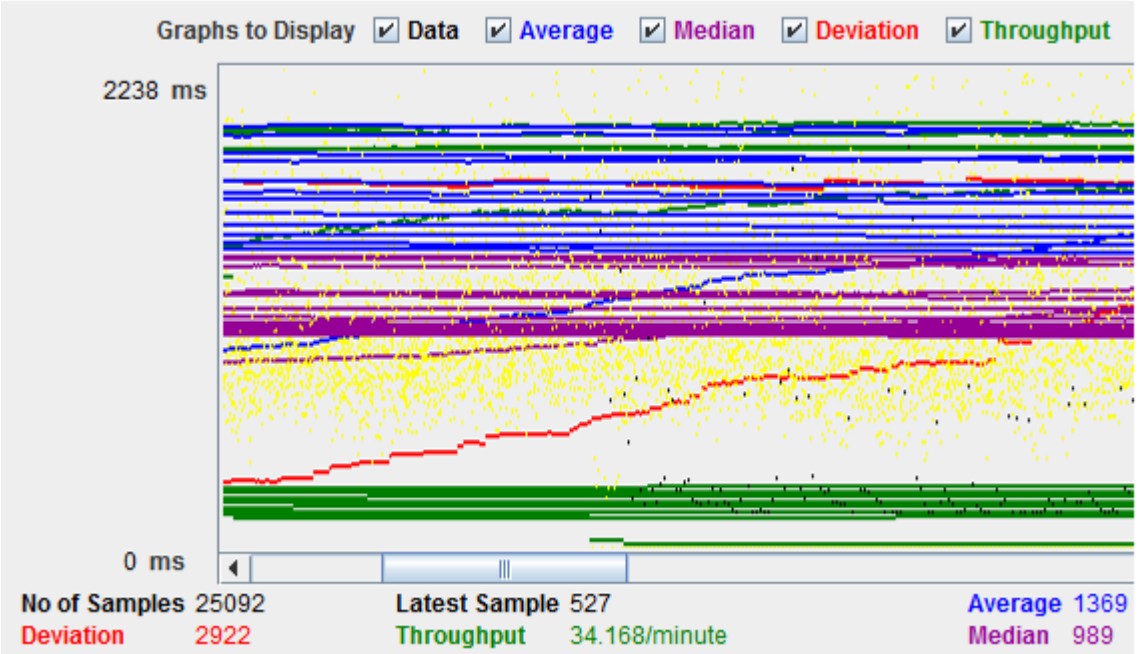
对于 JDBC Request、JMS Point-to-Point、JSR223、SMTP、JUnit Request 等 Sampler 组件元素就不一一说明了在后续的分享中，主要基于 HTTP 和 java 请求来分享实战。

4.4 Listeners(监听器)

在 jmeter 中 Listeners 提供了执行结果生成和显示能力的支持，提供了树形结构、表、图形和日志方式。

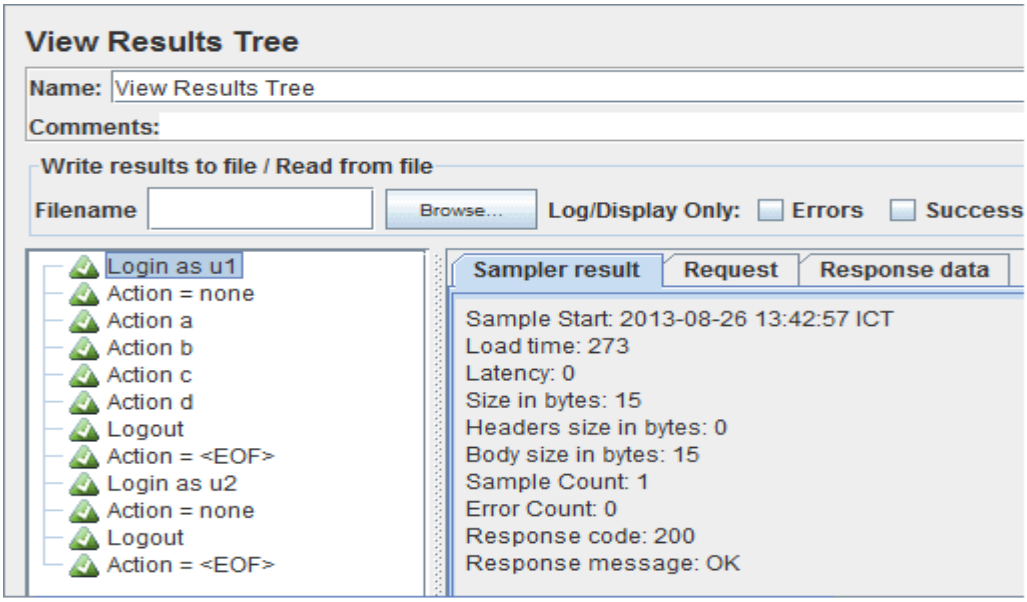
下面我们先看下几种结果显示示例图。

图形模式：



48.png

树模式：



49.png

表模式：

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename Browse... Log/D

Sample #	Start Time	Thread Name	Label	Sample Time(ms)
1	11:20:29.282	Thread Group 1-1	HTTP Request	1430
2	11:20:31.714	Thread Group 1-1	HTTP Request	1490
3	11:20:34.206	Thread Group 1-1	HTTP Request	534
4	11:20:35.743	Thread Group 1-1	HTTP Request	1966
5	11:20:38.714	Thread Group 1-1	HTTP Request	1247
6	11:20:40.964	Thread Group 1-1	HTTP Request	1140
7	11:20:43.107	Thread Group 1-1	HTTP Request	1631
8	11:20:45.740	Thread Group 1-1	HTTP Request	683

410.png

日志方式

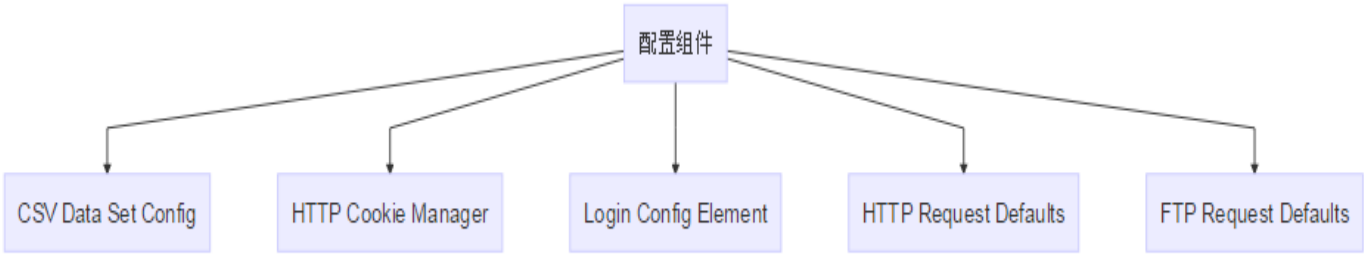
sampler_label	aggregate_	average	aggregate_
/v6exp3/redir.html	187	3517	185
/v6exp3/iframe.htm	168	1595	165
/first-android-testin	94	6009	1581
/search/adi/g.php	101	2999	21
/quality-center-tuto	67	3292	146
/b	41	3220	119
/bn/at_300.html	12	2174	1108
/getting-started-wit	8	2115	872
/sql.html	1	908	908
TOTAL	679	3225	21

411.png

4.5 Configuration Elements(即配置元件)

配置元件包含了 Samplers 下各种 Sampler 的默认配置设置，如果有配置默认配置，在 Sampler 下对应的 sampler 就会使用该默认配置。

下面我们看看我们主要用到的默认配置有哪些。

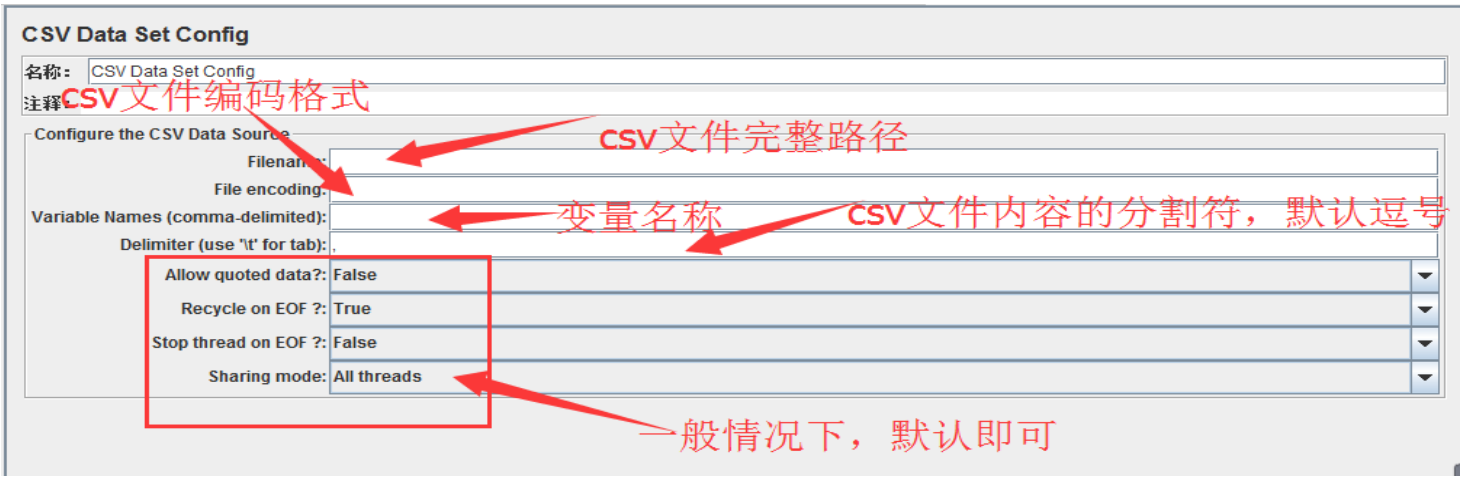


417.png

下面进行逐一的说明。

- CSV Data Set Config

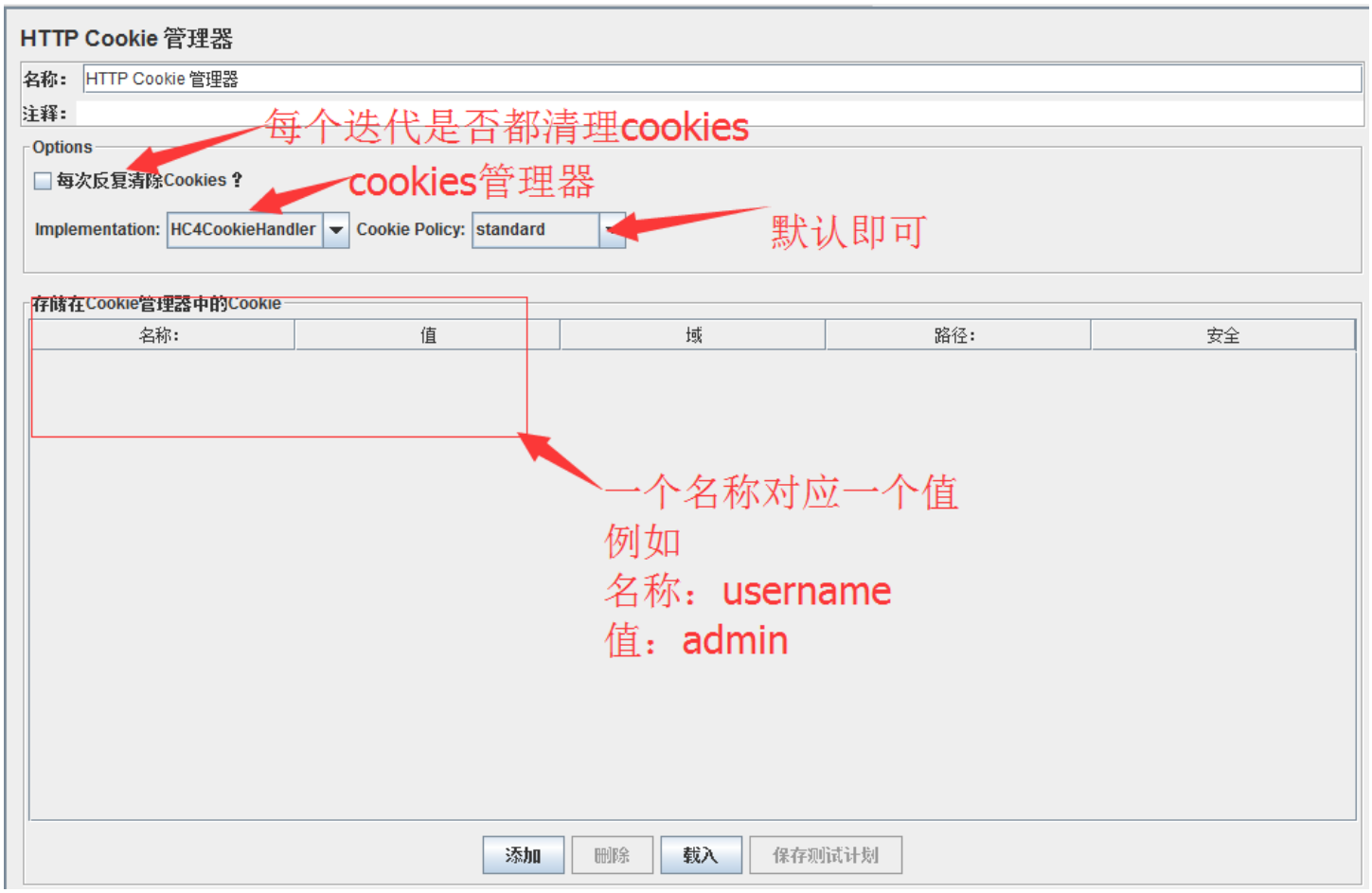
  CSV Data Set Config 主要用于读取 csv 格式的文件中数据，实现参数化。



413.png

- HTTP Cookie Manager

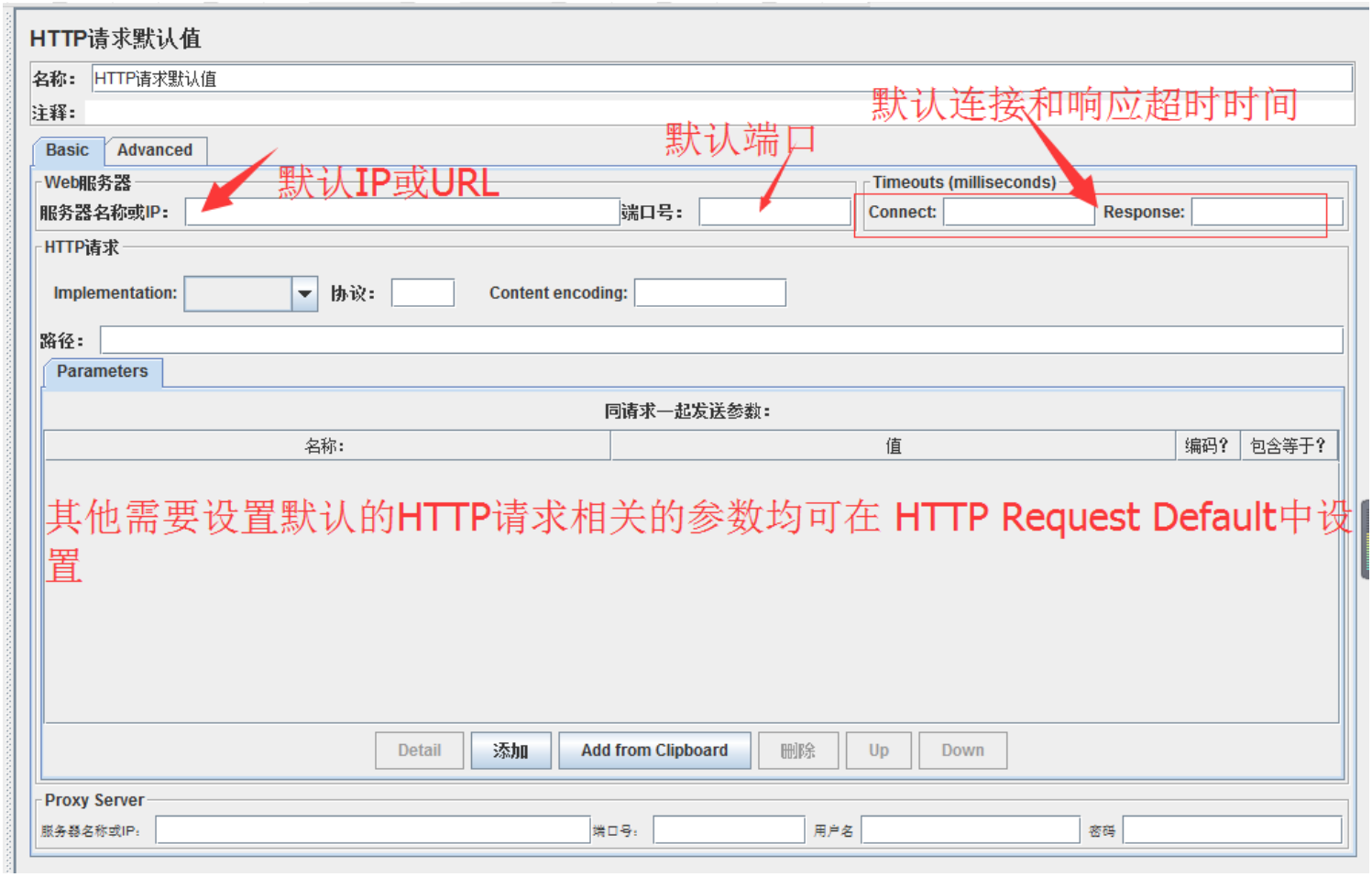
  HTTP Cookie Manager 主要用于默认 cookie 管理。



414.png

- HTTP Request Defaults

  HTTP Request Defaults 用于配置 HTTP request 的默认值，例如 IP、端口等等都设置好默认值后，在后续 HTTP request 元素里就不需要重复设置，节省时间。



415.png

4.6 总结

本次就 jmeter 常用的相关组件元素进行了大概的说明，以便大家有个基本的了解，为后续深入学习和实践打下基础。

第五篇 JMeter 性能测试基本过程及示例

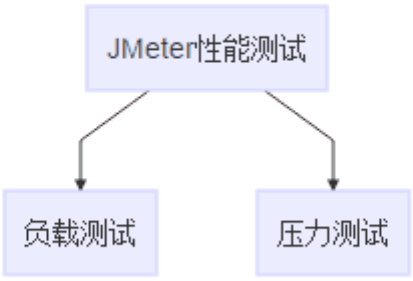
5.1 前言

性能测试是我们日常测试过程中，必须掌握的技能。通过进行性能测试，我们能分析服务端的整体性能、负载等，以便进一步评估我们的业务系统是否能满足当前运营生产及未来业务增长情况下如何进一步调整我们的服务配置方案。

jmeter 为性能测试提供了一下特色：

- jmeter 可以对测试静态资源（例如 js、html 等）以及动态资源（例如 php、jsp、ajax 等等）进行性能测试
- jmeter 可以挖掘出系统最大能处理的并发用户数
- jmeter 提供了一系列各种形式的性能分析报告

使用 jmeter 一般用于以下两种类型的性能测试



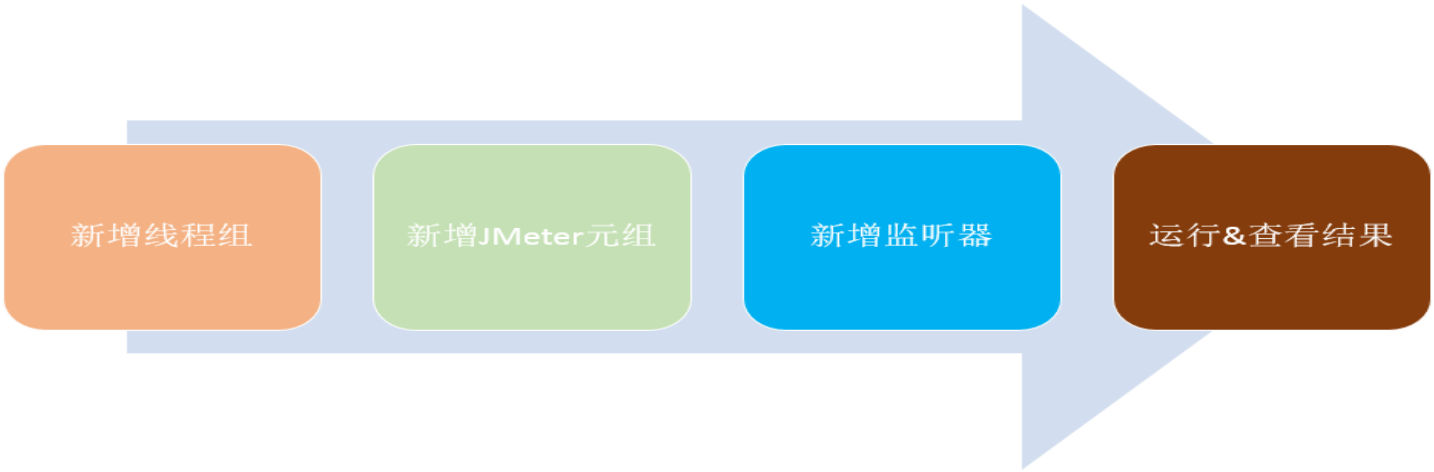
516.png

负载测试：通过测试系统在资源超负荷情况下的表现，以发现设计上的错误或验证系统的负载能力。

压力测试：测试系统能承受的最大负载能力。目的在于发挖掘出目标服务系统可以处理的最大负载。

5.2 基本过程

下面我们看下使用 jmeter 进行性能测试的基本过程。



51.png

对上图进行简要的说明

- 新增线程组

创建测试线程组，并设置线程数量及线程初始化启动方式。

- 新增 JMeter 元组

创建各种默认元组及测试元组，填入目标测试静态资源请求和动态资源请求参数及数据。

- 新增监听器

创建各种形式的结果搜集元组，以便在运行过程及运行结束后搜集监控指标数据。

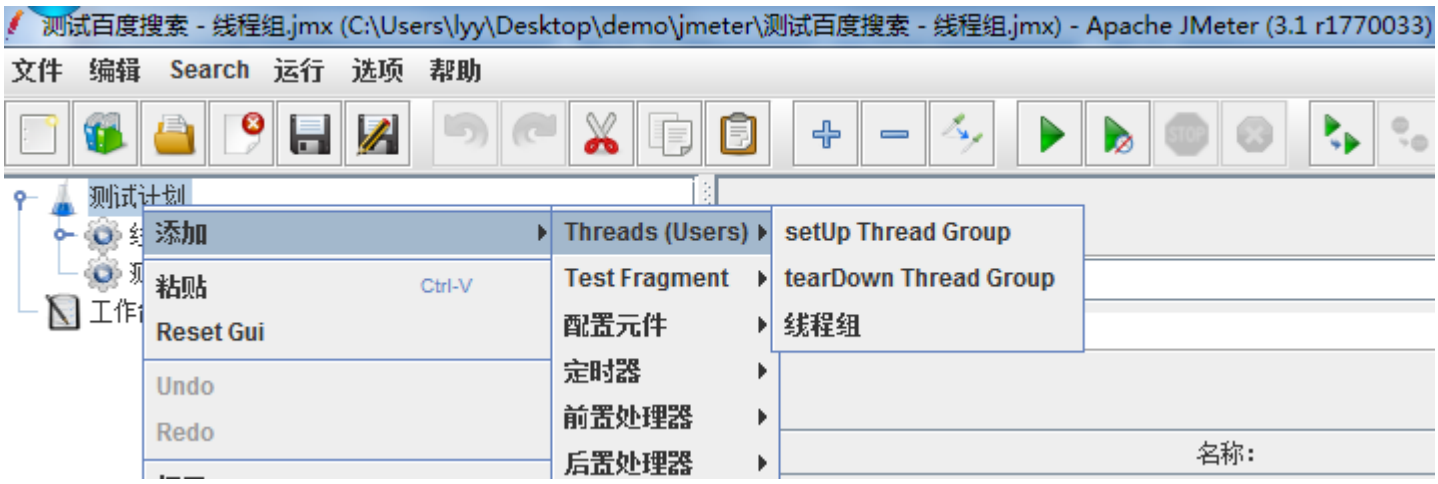
- 运行&查看结果

  调试运行，分析指标数据，挖掘性能瓶颈、评估系统性能状态、

5.3 示例

下面我们以打开百度演示上述过程。

- 新增线程组
- 在 jmeter 的 bin 目录下双击 jmeter.bat 启动 jmeter
- 在左边操作栏中选择“测试计划”，右击新增一个线程组，如图所示：

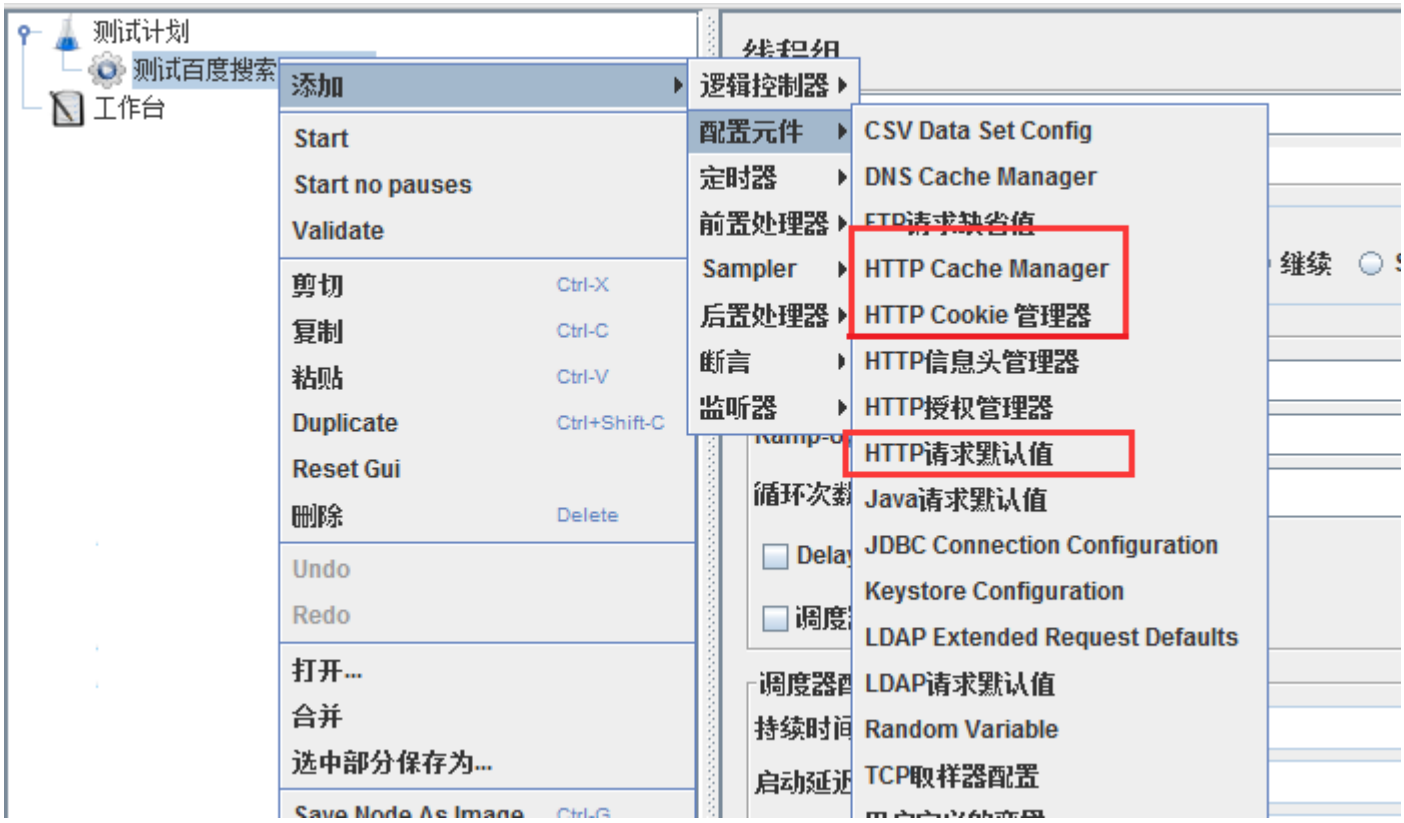


52.png

1. 初始化线程组相关信息，如图：

53.png

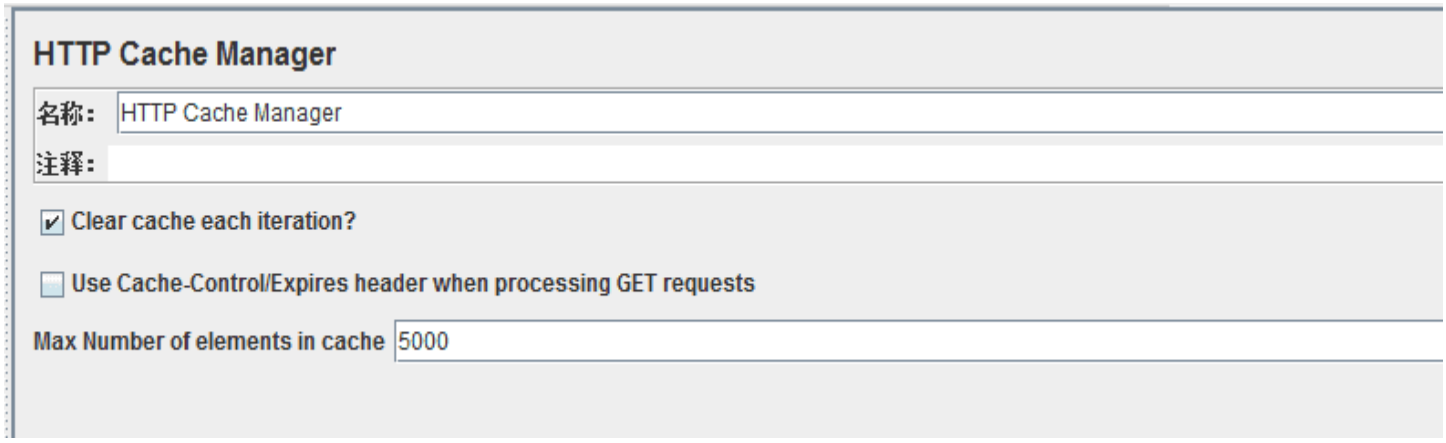
- 新增 JMeter 元组
- 添加默认配置元素，添加如下默认配置，如图



54.png

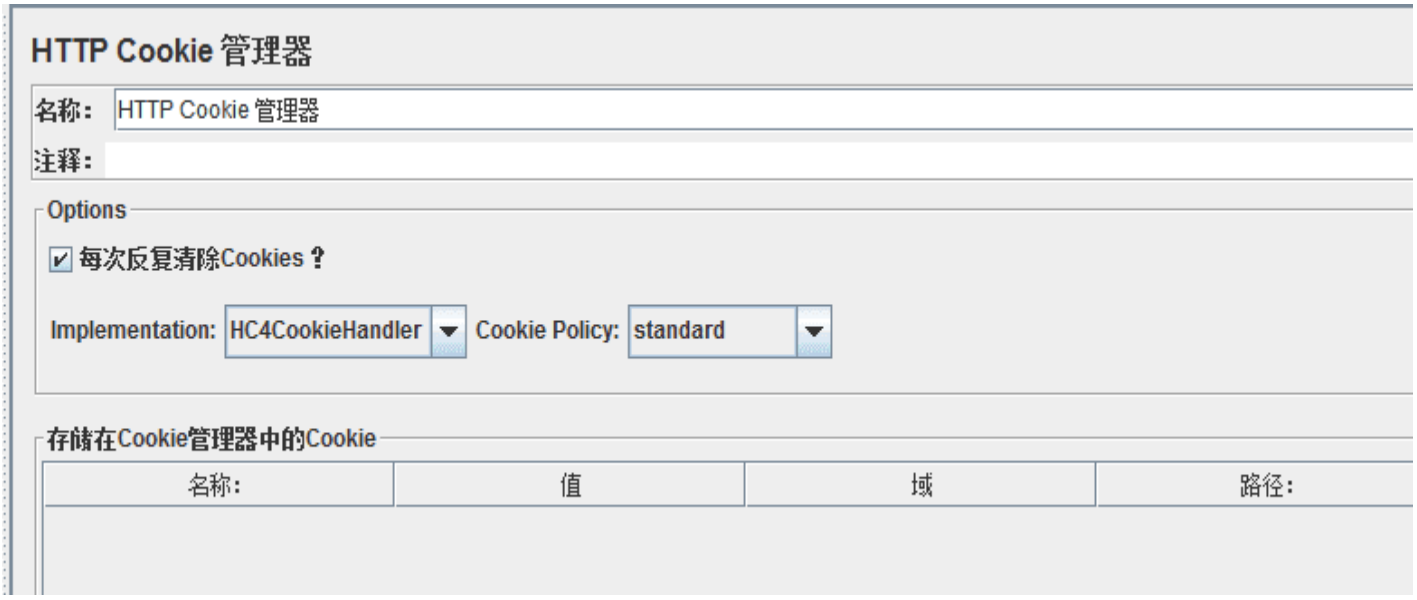
各默认组件配置如图所示。

HTTP Cache Manager



55.png

HTTP Cookie 管理器



56.png

HTTP 请求默认值

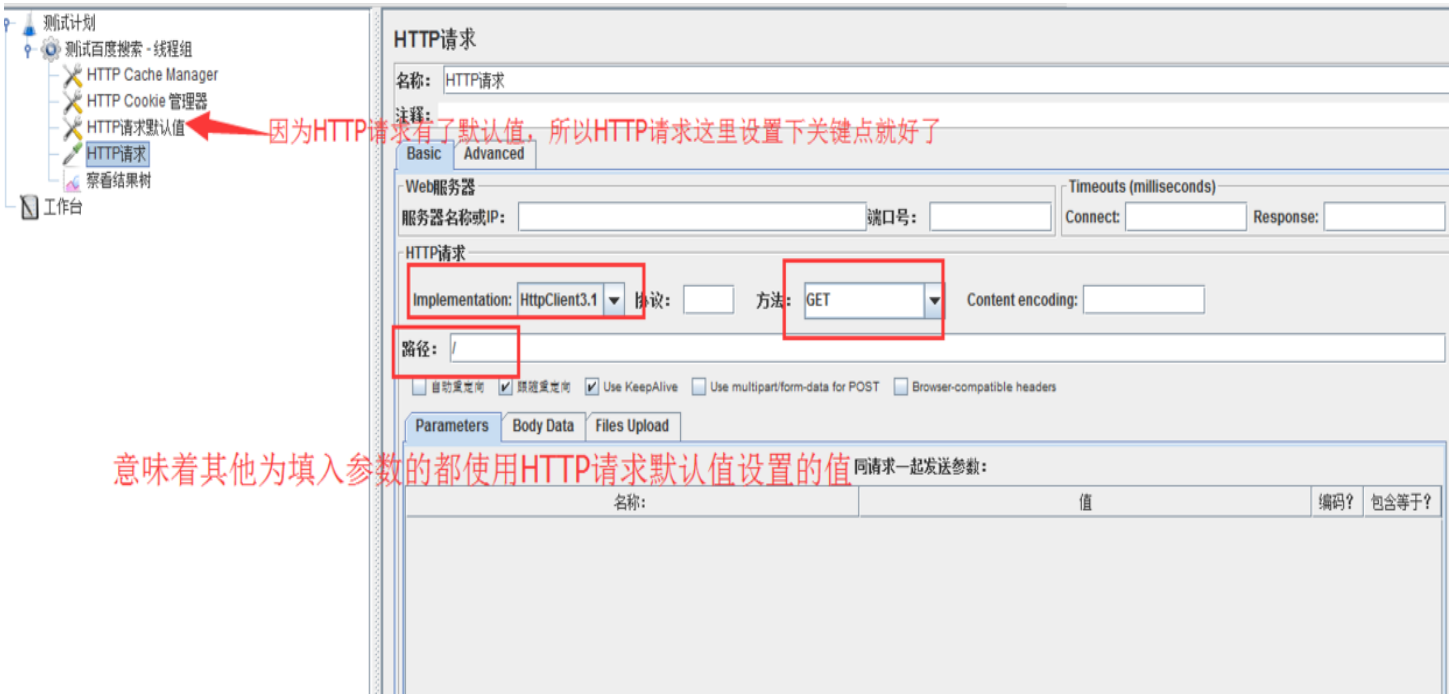
57.png

1. 添加 HTTP Request 元组

   在线程组上右击新增 HTTP 请求，如图：

58.png

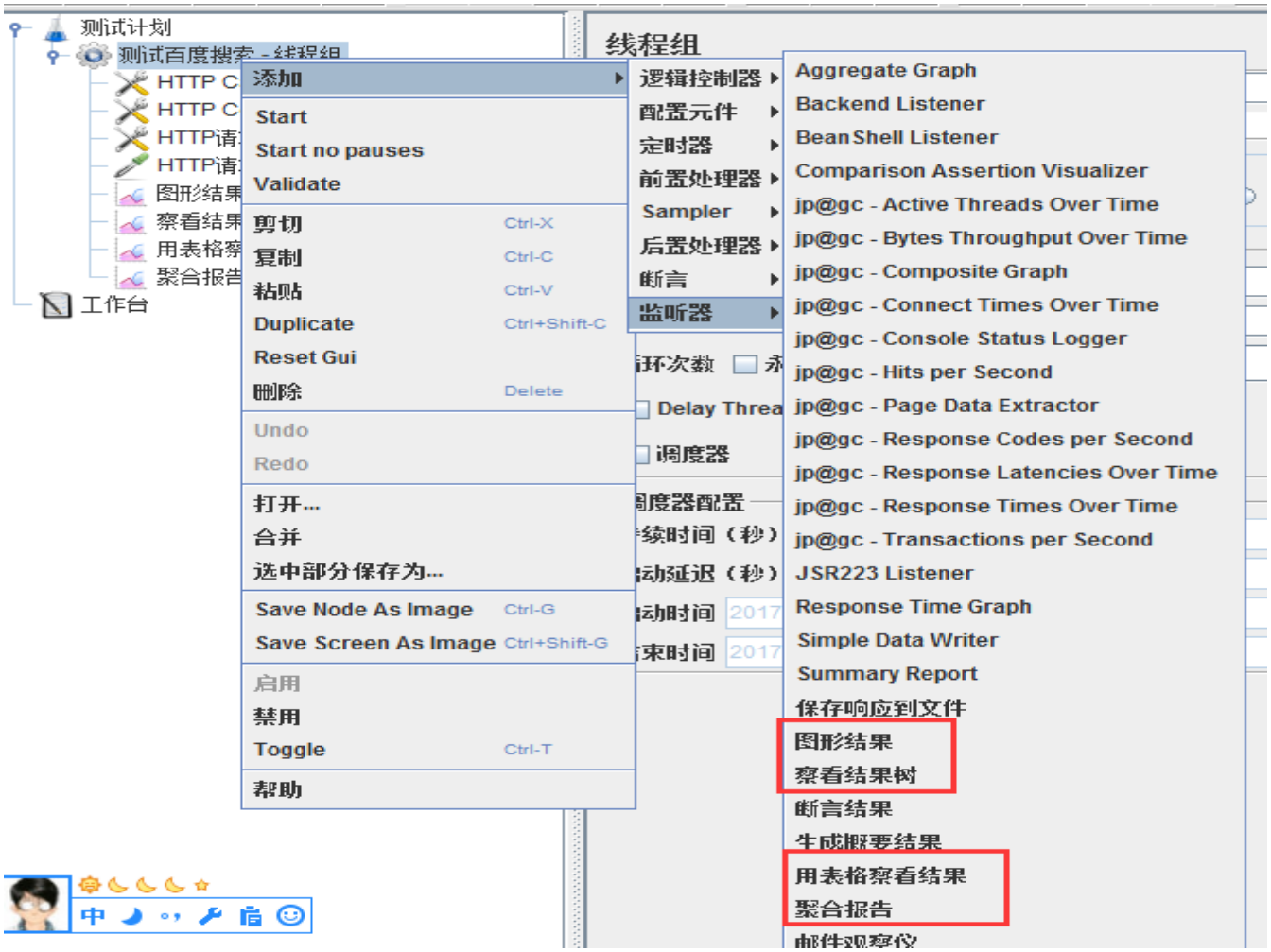
HTTP 请求设置如图：



59.png

- 新增监听器

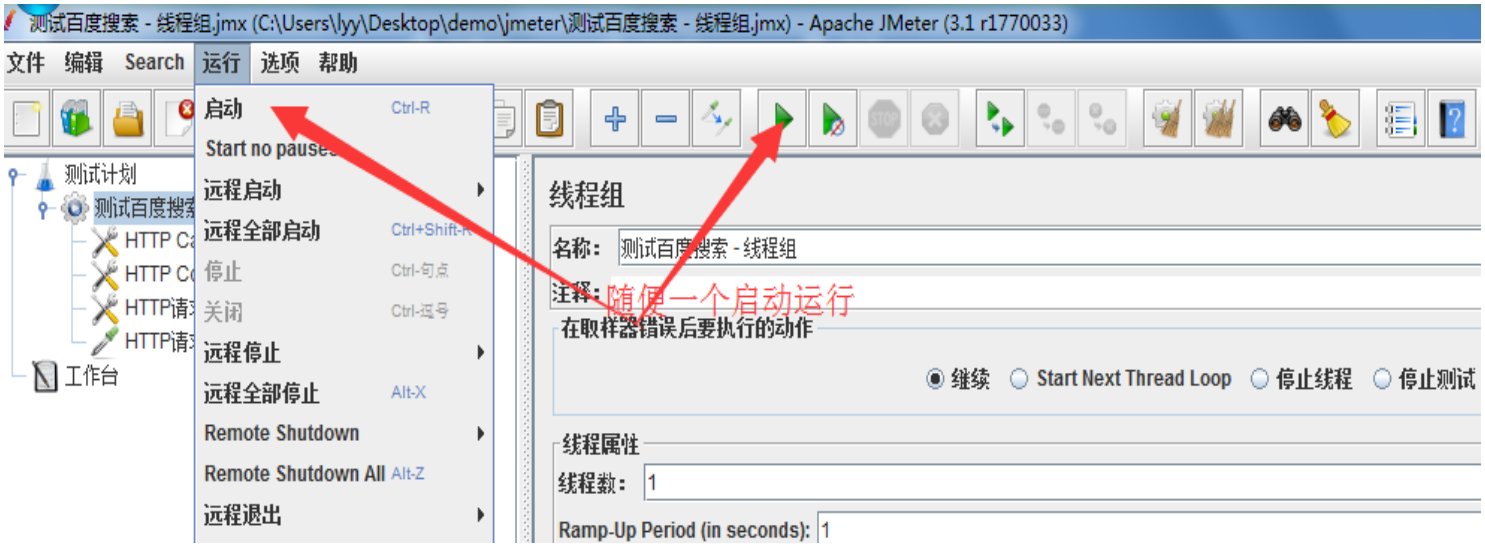
在这里我们添加如下监听器，如图所示



510.png

- 运行&查看结果

如果启动运行 jmeter，可以单击添加的监听器查看运行过程中的监控指标数据，也可以等运行结束后，再查看。

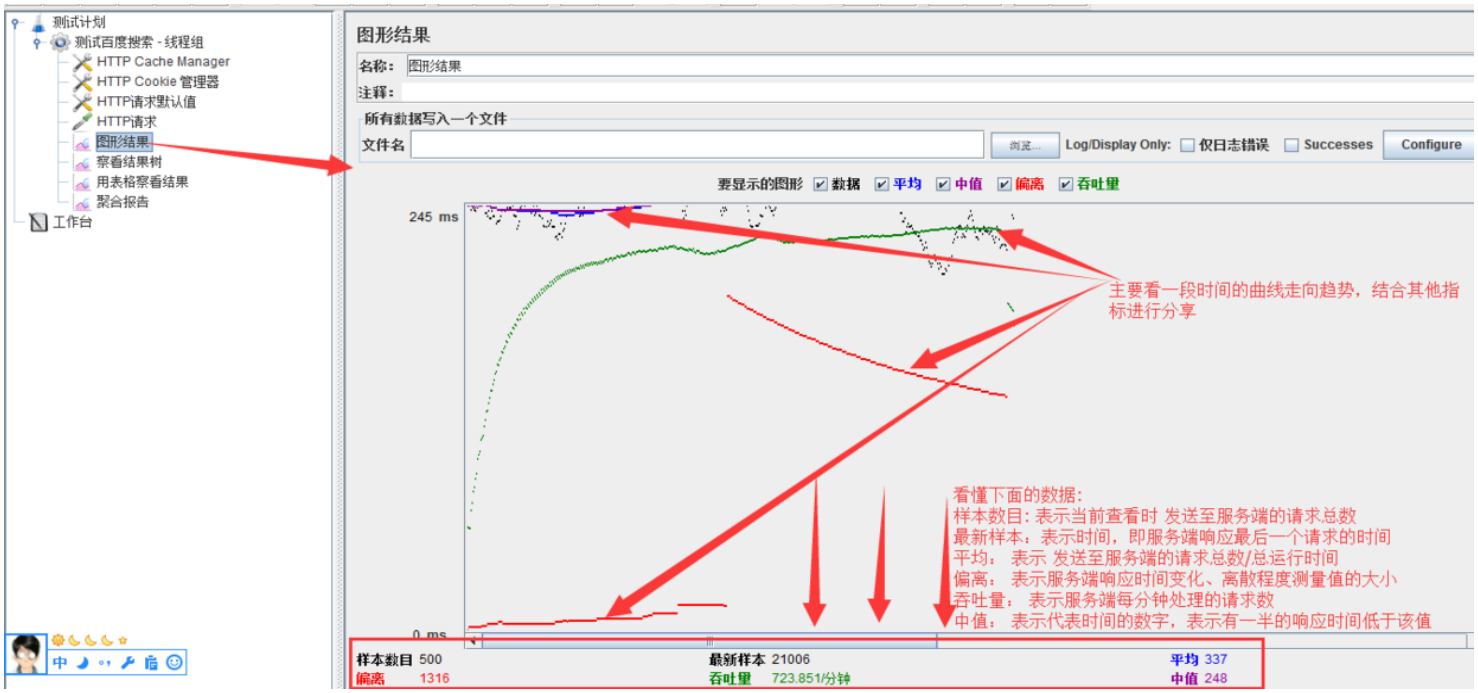


511.png

5.4 结果说明

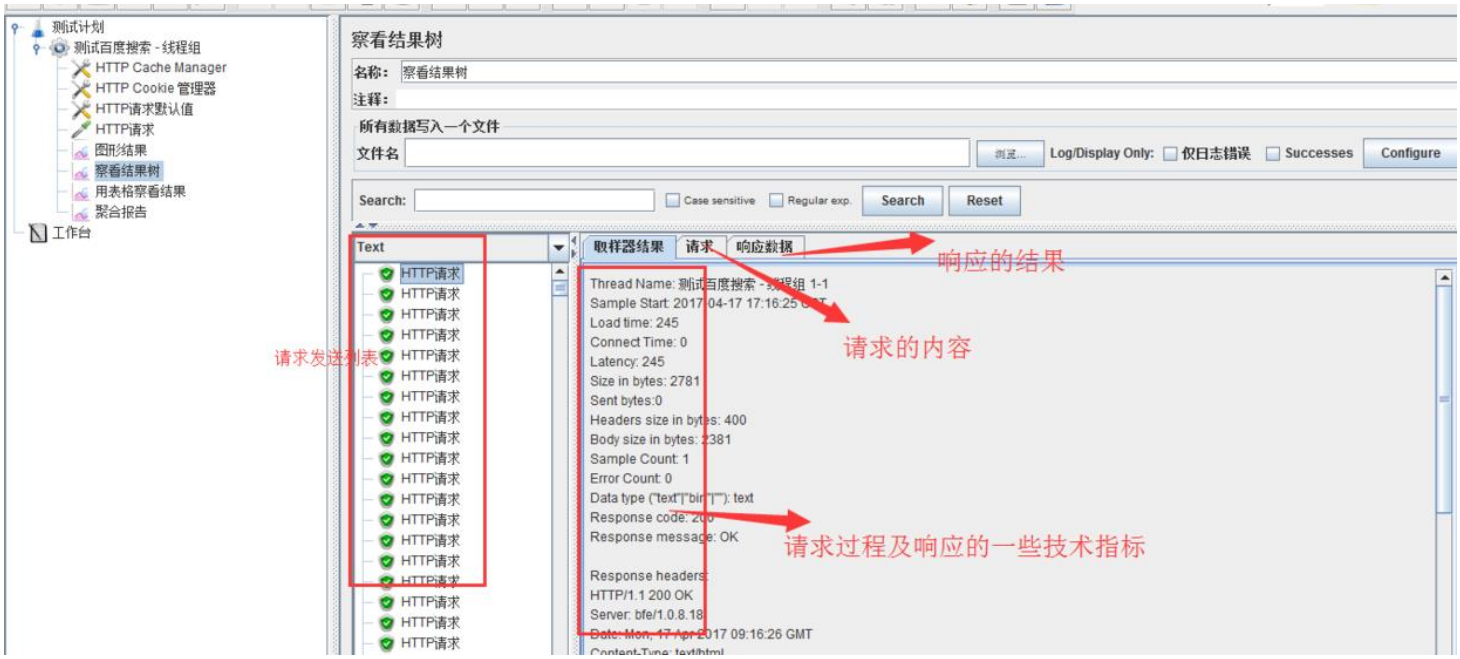
下面我们就监听器所采集的结果图进行简要的说明：

图形结果



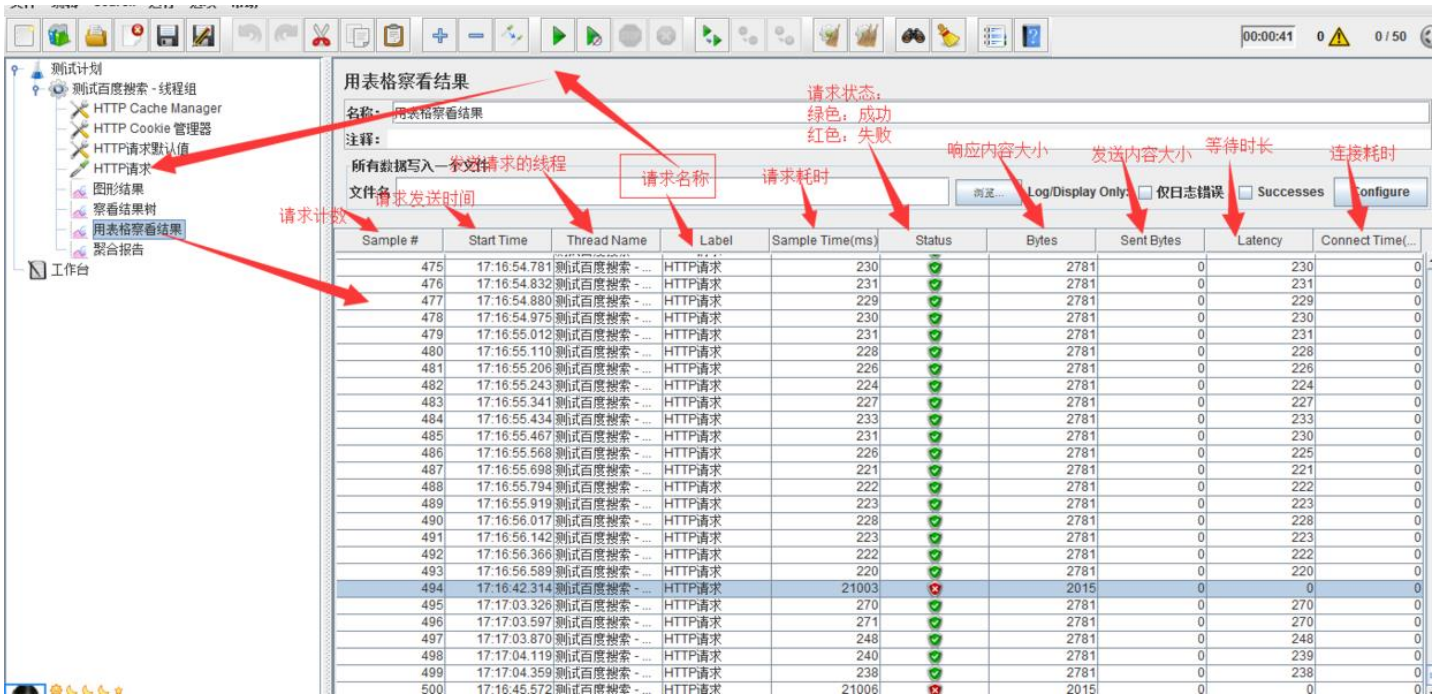
512.png

察看结果树



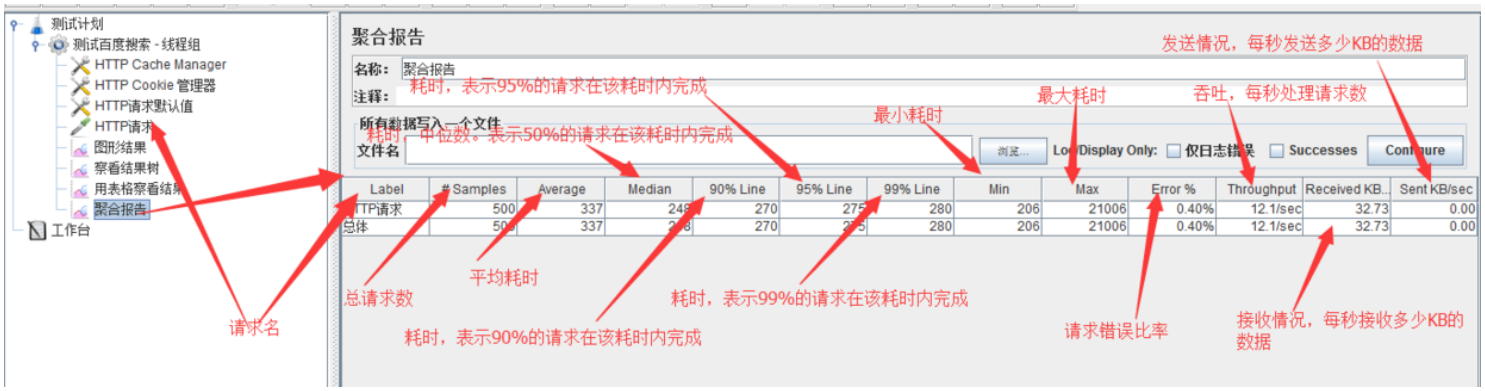
513.png

用表格查看结果



514.png

聚合报告



515.png

5.5 总结

本次就 jmeter 使用的基本过程如何使用进行了分享，并就访问百度首页进行了实际测试演示。在最后就常用的几个监听器中字段含义进行了说明。请大家根据企业实际项目进行演练，请勿使用示例中百度示例。

第六篇 JMeter 定时器

6.1 前言

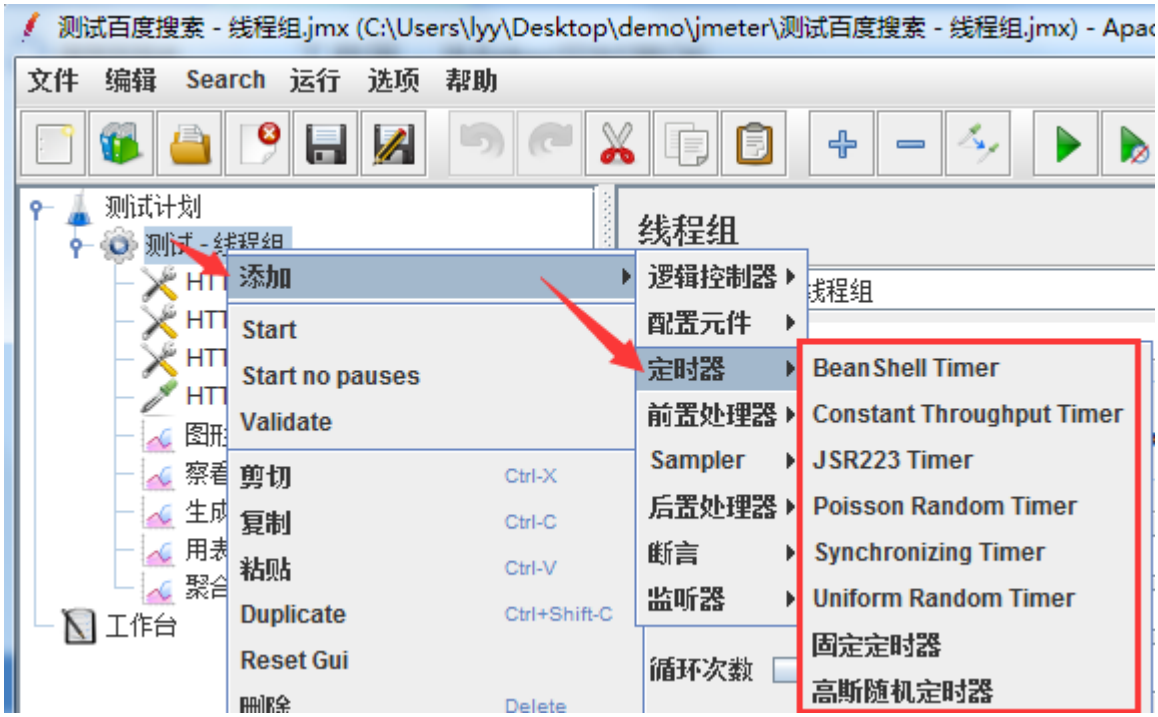
在默认情况下，jmeter 发送每个请求之间是没有延时的，如果采用默认方式，如果线程数足够大，瞬间就会将服务器压死。再则在实际的业务过程中，请求之间是有一定时间的停顿的

所以在请求之间设置合理的延时是必须的，也是更接近用户真实业务情况。

在 jmeter 中，定时器组件提供了系列不同类型的延时控制。合理使用定时器组件，能让你的性能测试更接近真实，更能挖掘出系统的瓶颈和评估系统的性能指标。

6.2 定时器类型

下面我们看下 jmeter 提供了哪些定时器组件：



61.png

- 固定定时器
- 高斯随机定时器
- Uniform Random Timer
- Synchronizing Timer
- Poisson Random Timer
- JSR223 Timer
- Constant Throughput Timer
- BeanShell Timer

6.3 固定定时器

这是最简单的一种定时器，也是新手最常用的一种方式。下面我们看下其具体设置：

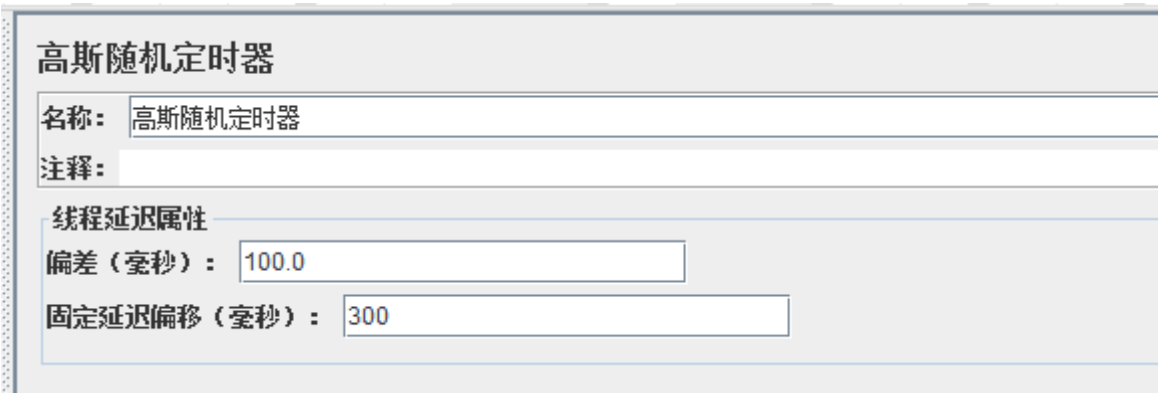
62.png

因其是固定值，在实际模拟用户请求的过程中，会失去灵活性，不推荐大量使用该定时器。

6.4 高斯随机定时器

高斯随机定时器，又可以称作正态分布随机定时器，该定时器可以设置在两个请求间随机延时时长。且总的延时是高斯分布(正态分布)的总和（均值：0.0、标准差 1.0）。在使用时须指定偏差延时值和偏移值。

。下面我们看下其具体设置：



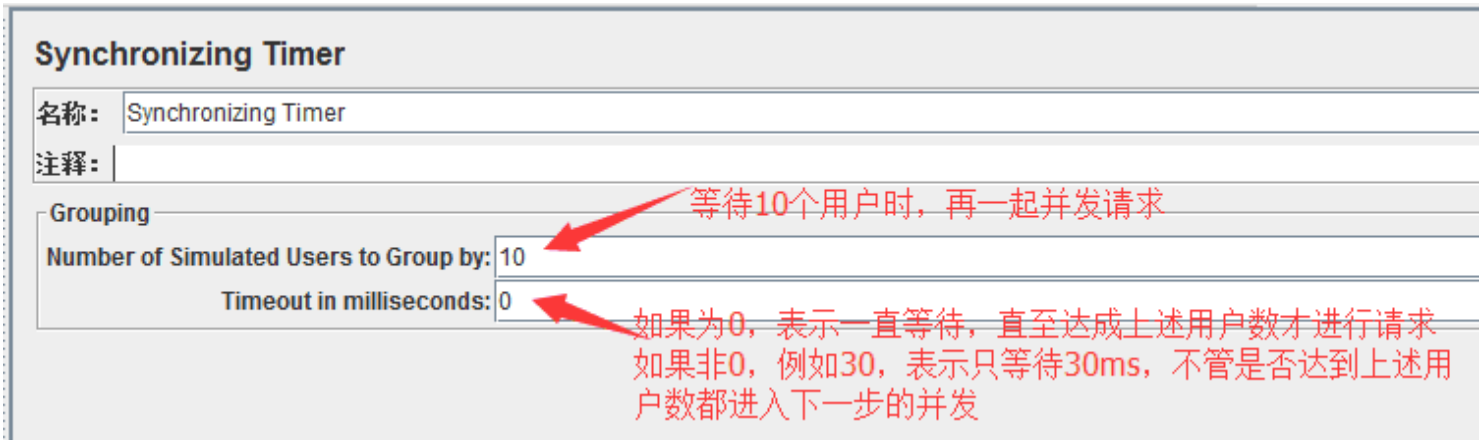
63.png

例如在访问百度首页，然后输入关键词进行搜索，受网络、人等各种因素影响，有的人打开首页后 3s 后则进行了搜索，有时则是 10s 或更多时间，在正常情况下，打开百度然后进行搜索，假设用户间隔在 3s-10s 之间，从统计学来看，这个间隔时间可能是一个正态分布或接近正态分布。而不是一个固定的常量。

从笔者在日常实践中，也更推荐使用该定时器。能更接近模拟用户实际情况。

6.5 Synchronizing Timer

这个定时器应该是大家很期望的，它有在 LoadRunner 中有一个大家熟悉的名称：集合点。是的，它实现了某种意义上的并发。

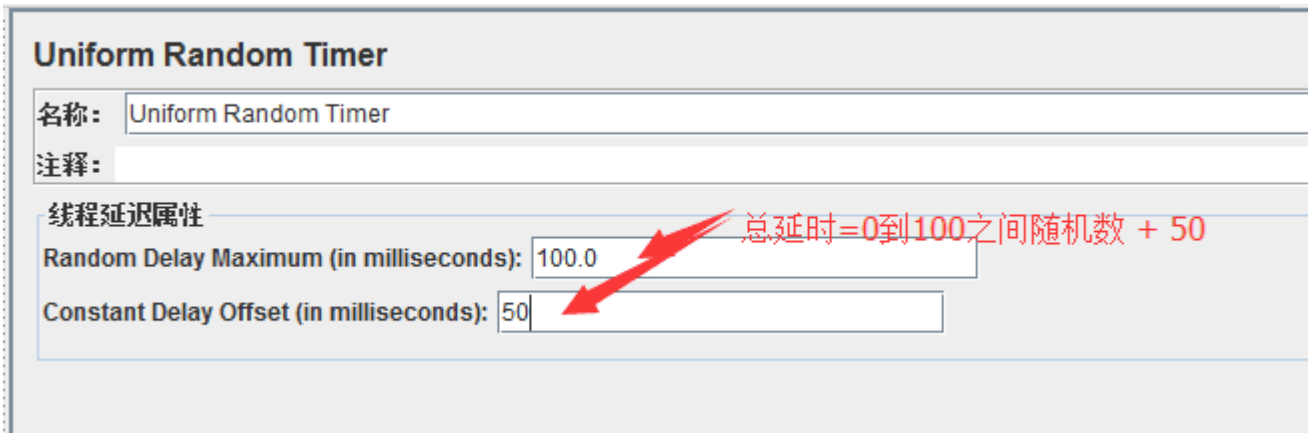


65.png

请注意 Timeout in milliseconds 尽量填写一个合理的值。

6.6 Uniform Random Timer

该定时器可以在请求之间设置一个随机延时，每个随机延时有相同的发生概率。总的延时等于随机延时 + 偏移延时值。

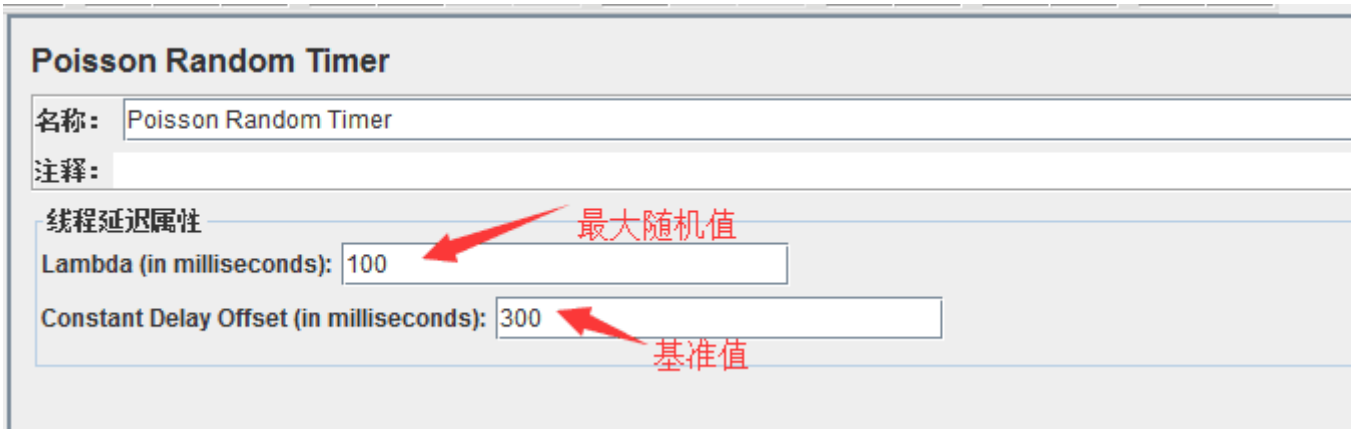


64.png

该定时器也是常用之一。

6.7 Poisson Random Timer

类似高斯随机定时器，只是其随机延时值发生在一个特定的值。总的延时值呈现泊松分布。



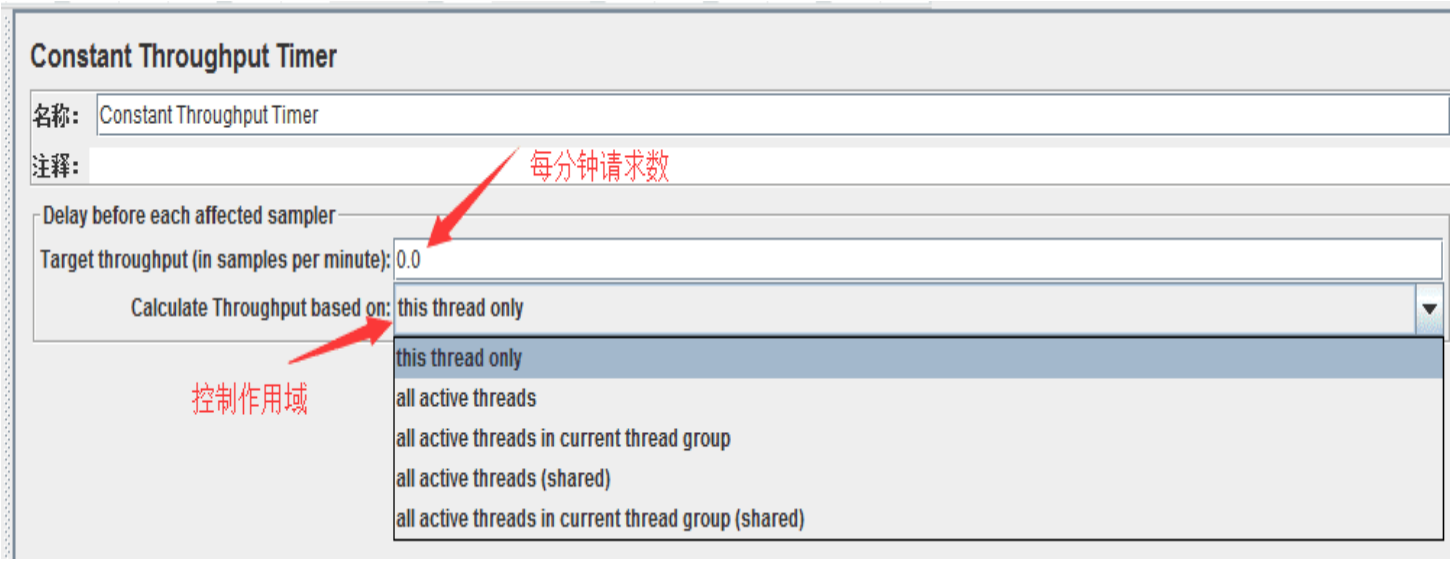
66.png

6.8 Constant Throughput Timer

通过控制每分钟请求数（即控制吞吐的方式）来控制是否进行延时暂停。

例如，当我们需要使服务端长期处于一定的压力下时，可以通过该定时器来控制吞吐。

注意：吞吐值可以是常量，也可以使用函数来动态生成，已达成更灵活的使用，满足不同的压力场景。



67.png

6.9 JSR223 Timer 和 BeanShell Timer

这两种定时器就不细说了，简单的说就是提供了脚本方式来进行控制，是更为灵活的方式。一般情况下，大家是不会用的。

当然有兴趣的，可以去研究下，增强理解。

6.10 总结

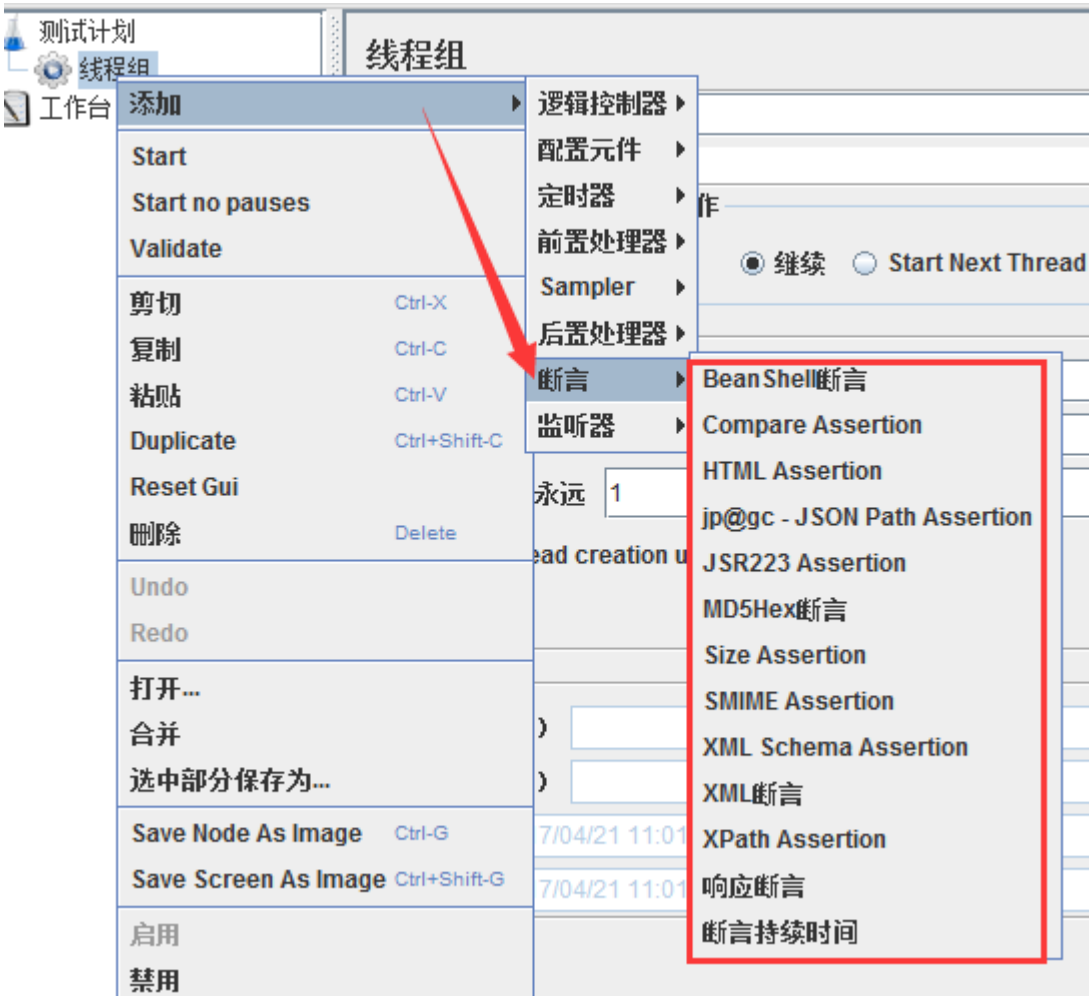
本文就各种定时器进行了介绍，并大致介绍了其可能的应用场景。不管是哪种定时器，都需要深入理解业务的情况下，统筹规划使用。以更深入的发挥其作用，模拟好真实应用场景，更好的挖掘性能瓶颈和评估目标服务的性能情况。

第七篇 JMeter 断言

7.1 前言

在 jmeter 中断言用于验证服务器返回的数据是否满足我们的要求。

jmeter 提供了以下断言类型：



1.png

下面我们主要对响应断言、XPath Assertion、jp@gc - JSON Path Assertion 进行分享，这几个断言类型也是日常压测过程中最常用的，对于其他的断言类型，请大家去看官方文档。

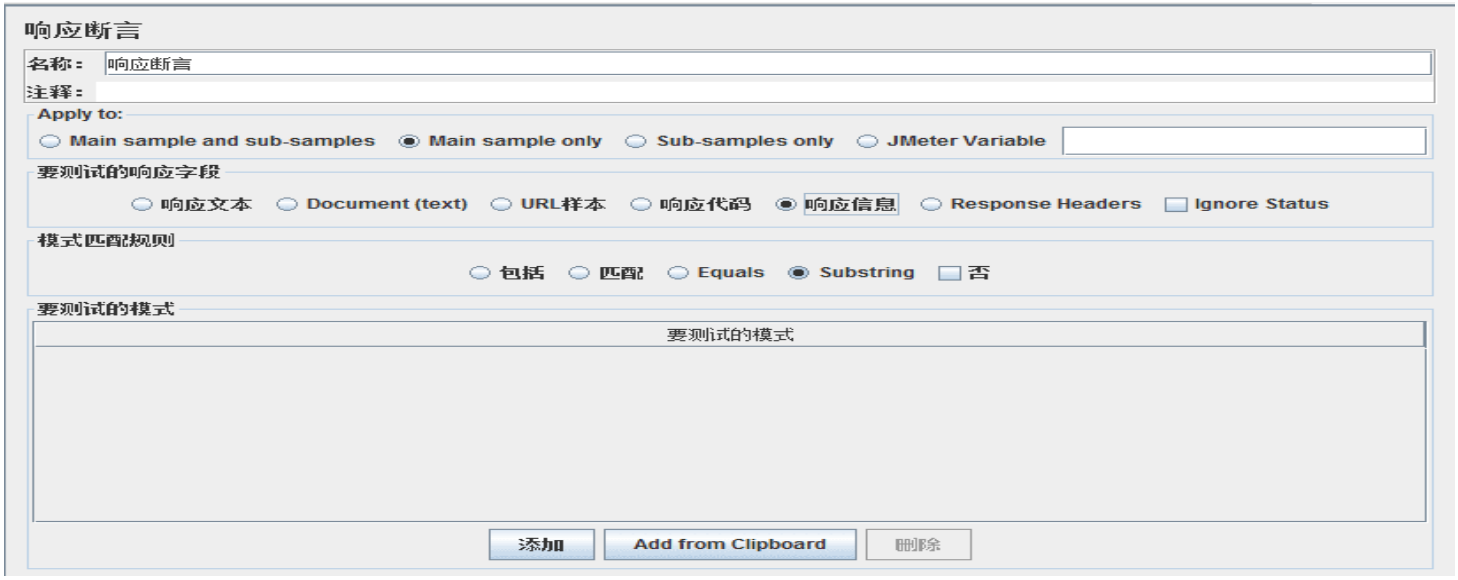
jmeter 提供了多大十几种断言方式，但合理利用好常用的几种断言就足以在驰骋于实际的项目应用了。

7.2 响应断言

响应断言允许用户通过添加模式字符串来比较验证服务器返回的响应。

例如对响应返回的状态码进行验证，或是对响应返回的本文内容验证等等。

下面我们对响应断言进行详细的说明：



2.png

- 名称、注释

  这里根据你实际的需要填写即可。

- Apply to

  一般选择 Main sample only 即可。如果一次发送多个请求，则需要根据实际断言需要选择其他选项了。（例如一个 ajax 请求，会发送多个 GET 或 POST 时。）

- 要测试的响应字段

- 响应文本

  服务器响应文本，一般情况下，我们都是勾选改选项，用于验证服务器返回值。

- Document（text）

  通过 Apache Tika 从各种的文档中提取的文本进行验证，包括响应文本，pdf、word 等等各种格式。jmeter 会用 Apache Tika 去解析服务器响应内容，耗内存、也耗时间，解析易失败，尽量少用或不用。多用响应文本方式来进行断言验证

- URL 样本

  对请求的 url 进行断言，如果请求没有重定向(302)，那么该 url 即为请求的 url；如果有重定向（切跟随重定向），那么 url 则包含了请求 url 和重定向 url。

- 响应代码

  即 http 响应代码，例如 200，404 等等，需要注意：

由于 jmeter 默认情况下认为 4xx，5xx 时该请求失败，所以在断言这类响应代码时，需要同时勾选 Ingore Status，才能正常去做断言。

- 响应信息

  即响应代码对应的信息，例如 OK， Not Found 等等这类的。

如下常见类似是响应信息:

HTTP/1.1 200 Ok

HTTP/1.1 302 Found

Response Header： 响应头信息，例如

Server: Tengine

Date: Thu, 12 Mar 2015 09:43:52 GMT

Content-Type: text/html

Content-Length: 260

Connection: close

Location: http://www.baidu.com/404.html

1. Response Headers

   即 http 响应头信息，主要用于断言当响应头带有唯一或特定意义时。

2. Ingore Status

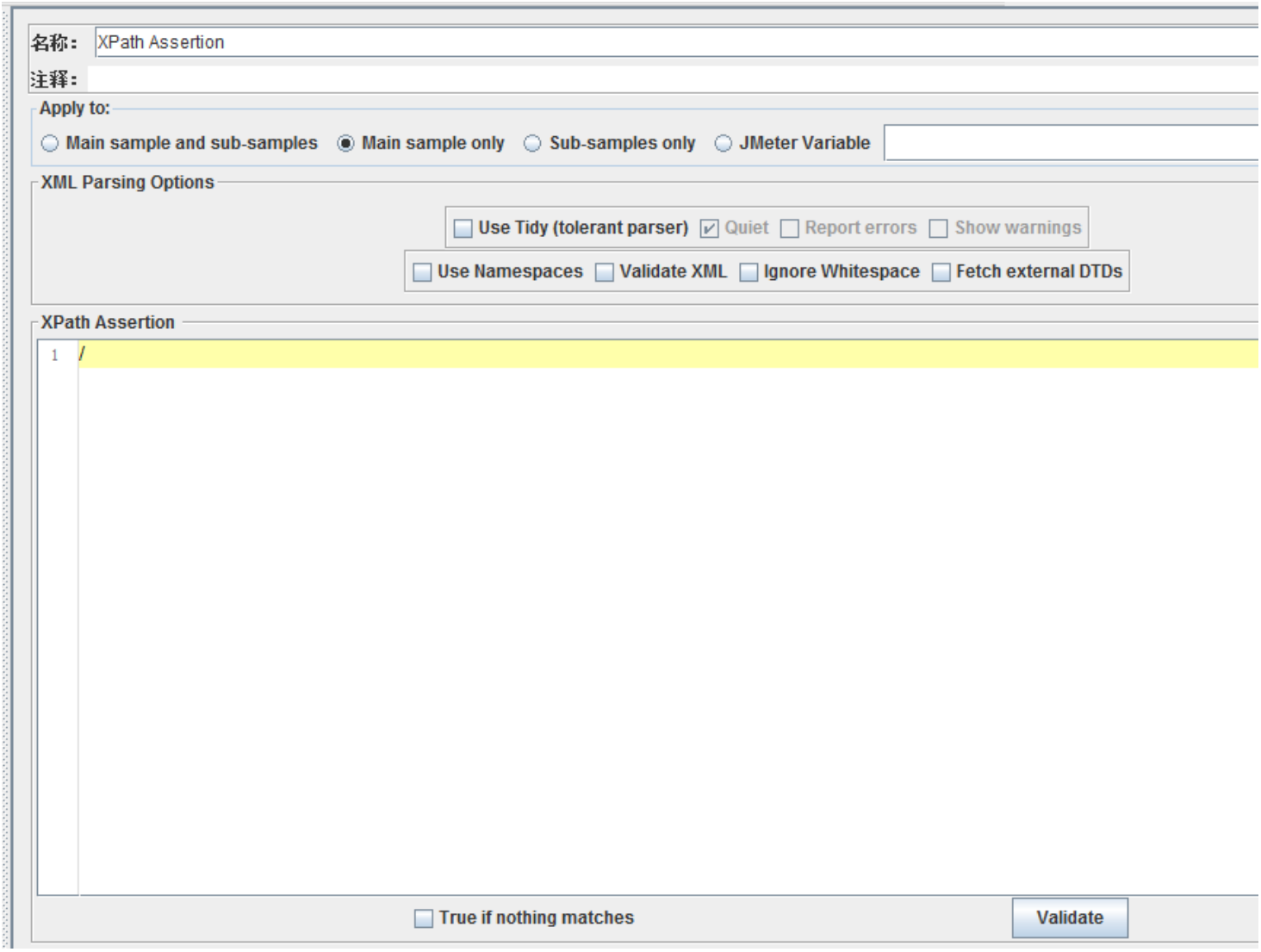
   请参见 4 响应代码的使用说明。

3. 模式匹配规则
4. 包括： 指返回结果包含要测试的模式中指定的内容，支持正则表达式
5. 匹配：（1）相当于 equals。返回值是固定的，可以以返回值做断言，效果同 equals；（2)正则表达式匹配。用正则表达式来匹配返回结果，但必须全部匹配。即正则表达式必须能匹配整个返回值，而不是返回部分值，注意与包括模式的区别（包括是支持模糊匹配的）。
6. Equals：指返回结果与指定的测试模式完全一致。
7. Substring：与“包括”模式差不多，都是指返回结果包括指定的内容，但 Substring 不支持正则表达式。
8. 否：相当于取反。即如果上述断言结果为 true，勾选“否”选项后，则最终断言结果为 false。

注：在使用该断言时，熟练掌握正则表达式是必备的能力。

7.3 XPath Assertion

如果服务器响应返回的是 xml 格式的内容，这时最佳的断言验证类型就是使用 XPath Assertion。



3.png

- Apply to

  一般选择 Main sample only 即可。如果一次发送多个请求，则需要根据实际断言需要选择其他选项了。（例如一个 ajax 请求，会发送多个 GET 或 POST 时。）

- XML Parsing Options

  Use Tidy(tolerant parser):使用 Tidy（容错解析器），默认选择 quiet

Quiet：不显示

Report errors：错误报告

Show warnings:显示错误

Use Namespaces:使用名称空间

Validate XML:验证 XML（文件包/数据）

Ignore Whitespace:忽略空格（允许你指定语法分析器可以忽略哪个空格，而哪个空格是重要的）

Fetch external DTDs:获取外部 DTDs（一些 XML 元素具有属性，属性包含应用程序使用的信息，属性仅在程序对元素进行读、写操作时，提供元素的额外信息，这时候需要在 DTDs 中声明）

- Path Assertion

  输入框中写入 xpath 断言，点击 Validate 验证其正确性

- True if nothing matches

  确认都不匹配

7.4 jp@gc - JSON Path Assertion

如果服务器响应返回的是 json 格式的内容，这时最佳的断言验证类型就是使用 jp@gc - JSON Path Assertion。

4.png

注： 默认下载的 jmeter 是不支持该方式的，需要安装 json plugins，在选项-Plugins Manager-Available Plugins 找到 JSON Plugins 安装好即可。

下面对 json path assertion 进行说明

- JSON Path

  json 提取表达式，用于提取目标 json 串节点值。
- Validate against expected value

  勾选该选项，则验证目标期望结果
- Match as regular expression

  勾选该选项，则期望值项，支持正则表达式

- Expected Value

  自定义期望值

- Expect null

  期望值为 null，勾选该选项，则会断言结果为 null 的情况

- Invert assertion(will fail if above condition met)

  取反，如果上述两种期望值断言为 true，勾选该选项，则断言结果为 fail；如果上述期望值断言为 fail，勾选该选项，则断言结果为 true。

7.5 总结

本次分享主要就响应断言、XPath 断言、JSON 断言三种常用的断言类型进行了说明，对于具体的示例，后续在实践篇章会结合其他基础功能一一进行分享，这三种断言应该说满足日常压测过程断言的大部分场景，大家需要深入理解其各个选项的含义。

第八篇 JMeter 逻辑控制器

8.1 前言

在 jmeter 中逻辑控制器主要分类两类：

- 控制 jmeter 测试计划中节点的逻辑执行顺序等等
- 对 jmeter 的节点进行分组，方便结果统计等等

进一步简化下，笔者把逻辑控制器分为

- 逻辑控制类
- 分组控制类

1.png

8.2 逻辑控制类

逻辑控制类控制器定义了在执行线程中请求的执行顺序。

下面我们就常用的逻辑控制器进行说明

8.2.1 如果(if)控制器

控制其下面的子节点满足条件才执行，例如，我们控制只有执行线程大于 10 个时，才执行其子节点。

2.png

这里只是简单举例，大家可以根据实际应用场景进行设计。

8.2.2 循环控制器

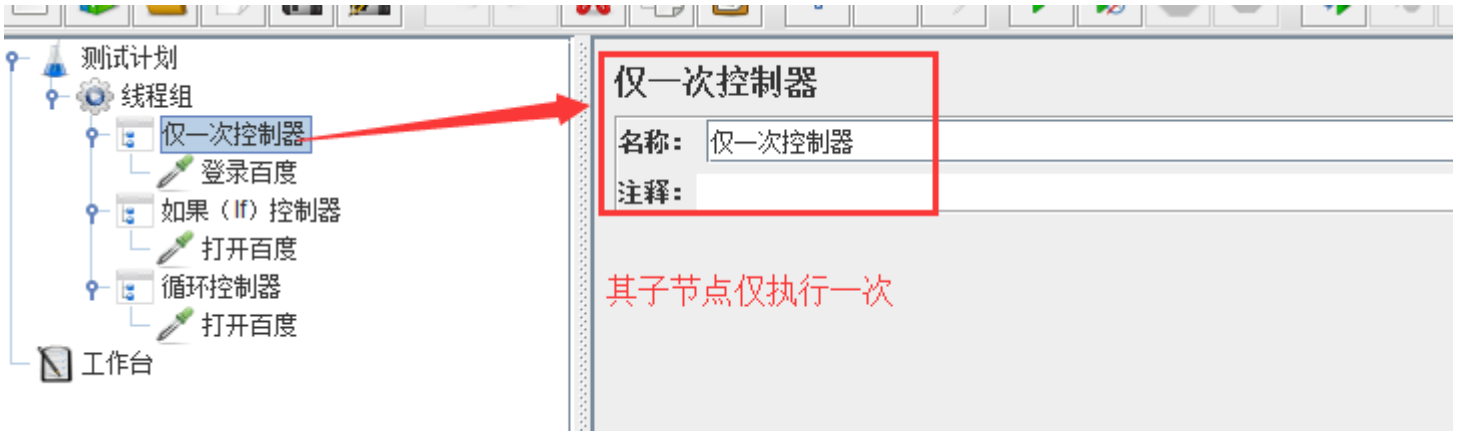
控制其下面的子节点运行次数。例如我们设置其子节点执行 10 次。

3.png

如果勾选永远选项，则会一直执行下去。

8.2.3 仅一次控制器

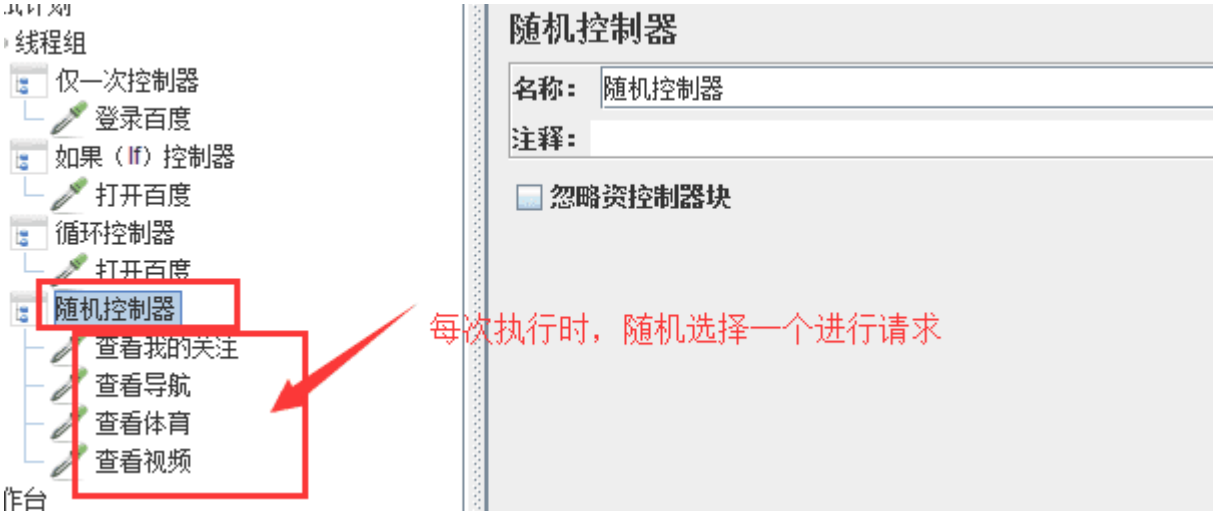
控制其子节点在整个测试计划执行期间的每个线程仅执行一次，例如我们可以用于等登录动作。



4.png

8.2.4 随机控制器

每次执行时，从其子节点中，随机选择一个进行执行，例如我们百度首页随机请求不同的类型的资讯信息。



5.png

其他的逻辑控制器就不一一进行说明了，大家可以自行学习、实践，去挖掘其实用场景。

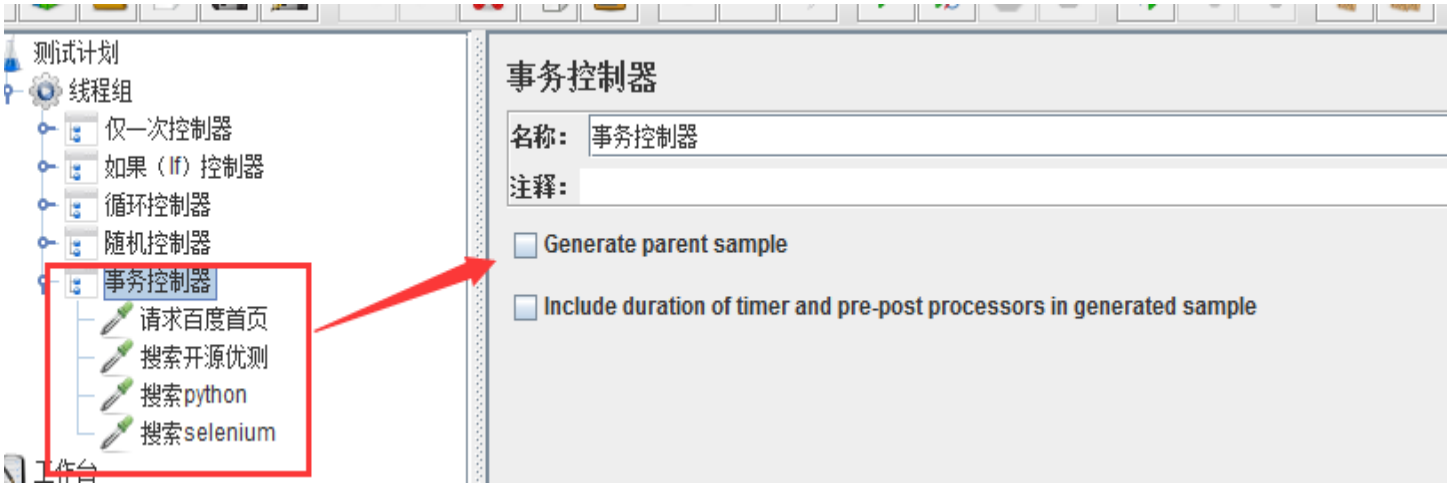
8.3 分组控制类

分组控制类主要用于统计和控制其他非逻辑执行。典型的应用场景，例如我们常需要去统计一个业务流的执行时间，或是控制吞吐量等等。

下面我们一起看几个典型的分组控制类的组件。

8.3.1 事务控制器

会产生一个额外的 sampler，用于统计该控制器下子节点的所有时间。该统计数据可以在聚合报告中看到。



6.png

Generate parent sample：控制结果的显示结构。若勾选，总时长和子节点时长按层级显示，未勾选，平行显示

Include duration of timer and pre-post processors in generated sampler：勾选时，会统计定时器时间（默认仅统计采样器时间）

如上图：通过事务控制器，我们可以统计出请求百度首页、搜索开源优测、搜索 python、搜索 selenium4 个请求的时间总和，注意这里统一出来的时间会略大于这 4 个请求的和。

8.3.2 吞吐控制器

允许用户通过以下两种方法控制执行频率。

7.png

- Percent executions

  这个控制器的命名不够准确，因为它不是用来控制吞吐量的。吞吐量控制器允许用户控制执行频率，jmeter 提供了两种模式：执行百分比和执行总次数。

  设置运行比例(1~100 之间)

  如线程循环次数设置为 5，添加 Percent executions 为 40%的吞吐量控制器，其下子节点则循环 2 次

- Total executions

  设置运行次数

  per user：此项被勾选后，在每个线程的基础上，每个用户都将根据控制器设置计算。未被勾选时，计算针对于所有用户。

  如：使用 total

execution 模式，不勾选 per user 选

项，执行次数=吞吐量值；勾选了 per user，执行次数=user 数量（对应线程数） * 吞吐量值

8.4 总结

本次就常用的逻辑控制器：如果(if)控制器、循环控制器、仅一次控制器、随机控制器、事务控制器、吞吐控制器进行了分享。对于这些控制器的应用场景，需要深刻理解业务后再设计场景，切不可为了用而用。

第九篇 JMeter 处理器

9.1 前言

在 jmeter 中提供了两种处理器，用于修改请求数据或处理响应数据。

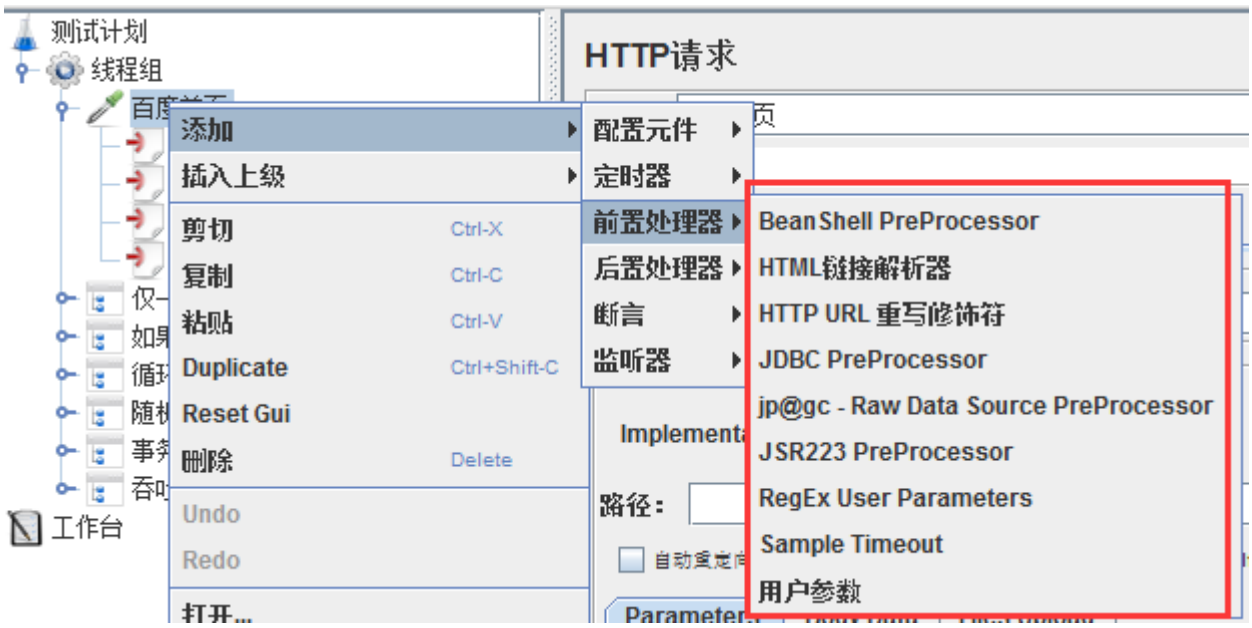
1.png

- 前置处理器
- 后置处理器

9.2 前置处理器

前置处理器是在请求发送前做相关处理。可以用于在请求发送前修改 HTTP 协议头、数据部分等等各种需要修改或设置的数据。

其作用范围内的每一个 sampler 元件之前执行。



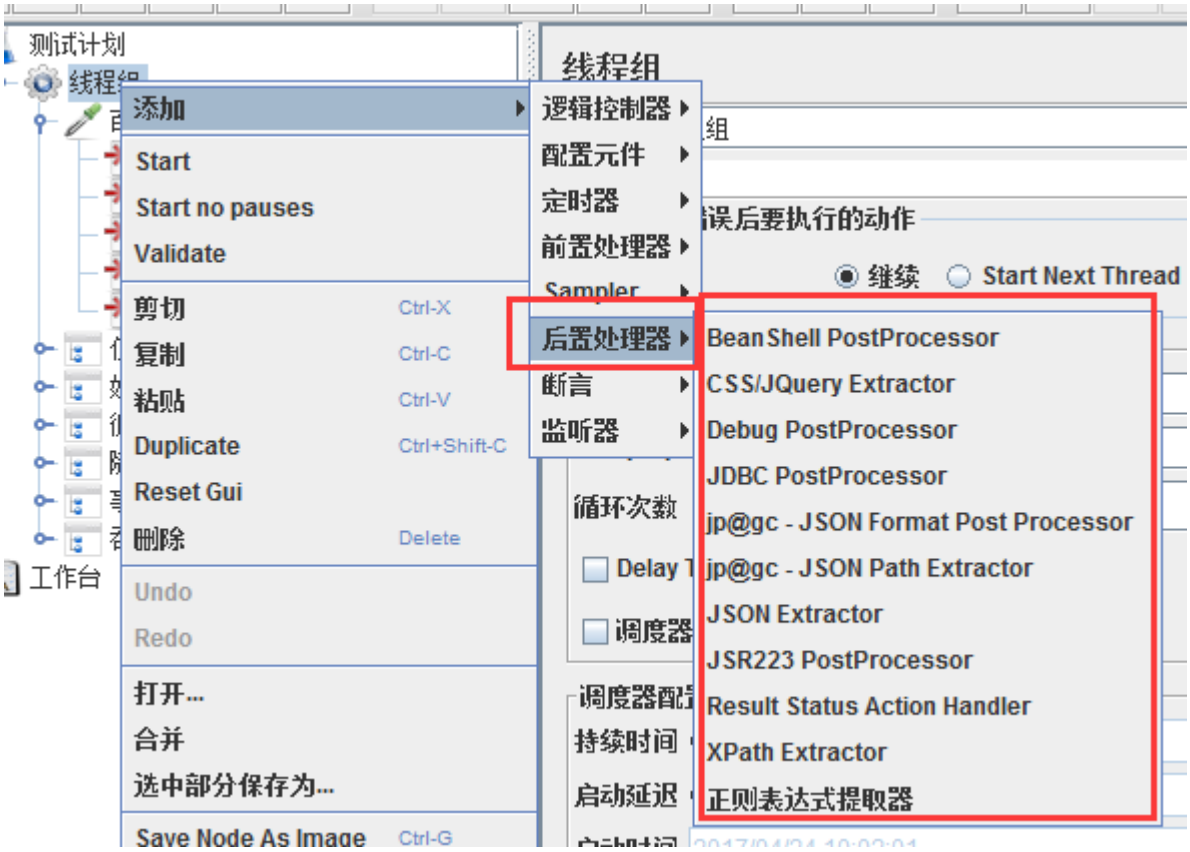
2.png

- Bean Shell PreProcessor
- HTML 链接解析器
- HTTP URL 重写修饰符
- JDBC PreProcessor
- jp@gc - Raw Data Source PreProcessor
- JSR223 PreProcessor
- RegEx User Parameters
- Sample Timeout
- 用户参数

注： 一般情况下，大家在实践过程中，用到前置处理器的机会比较少，这里就不一一说明了，重点放在后置处理器的讲解上。

9.3 后置处理器

后置处理器是取样器被执行后被触发执行的元素。可用于解析响应数据，提取变量，以便后续使用。

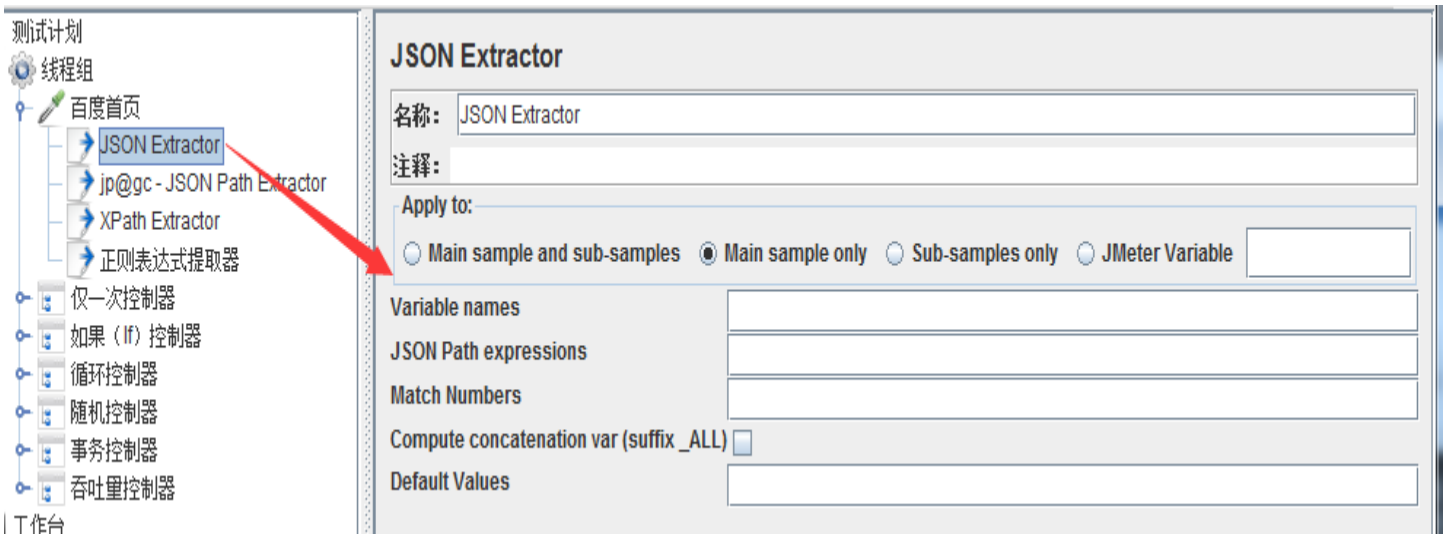


3.png

==注： json 格式的支持需要安装 json plugins 创建==

下面我们对常用的后置处理器进行说明：

- JSON Extractor



4.png

用于处理响应结果为 json 格式的内容。

Variable names：变量名称，提取到的值将存放在该变量里，后续通过该变量即可引用提取到的数据

JSONPath Expression：JSON 表达式

Match Numbers：匹配哪个，可为空即默认第一个

Default Value：未取到值的时候默认值

示例

例如返回的 json 串为，我们提取 token：

```
{
    "statusCode":200,
    "data":{
```

```
        "userId":"admin",
        "token":"123123123123338a5bd20bd"
    }
}
```

在 JSONPath Expression 填入：

```
$.data.token
```

来获取 token 的值

例如返回的 json 串有数组，我们提取第二个 token：

```
{
    "statusCode":200,
    "data":[{
        "userId":"admin",
        "token":"rwerwerwr0e6138a5bd20bd"
    },{
        "userId":"user",
        "token":"123123123123123a5bd20bd"
    }]
}
```

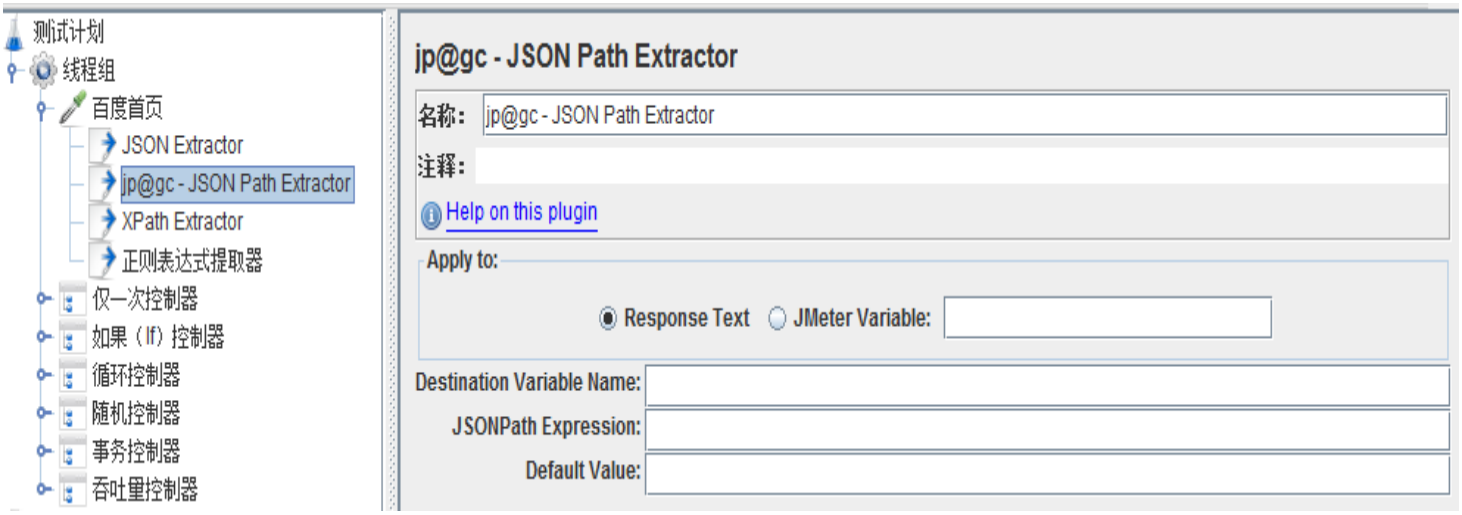
在 JSONPath Expression 填入：

$.data[1].token

来获取第二个 token 的值（注：数组的索引从 0 开始表示第一个）

- jp@gc - JSON Path Extractor

  用于处理响应结果为 json 格式的内容。



5.png

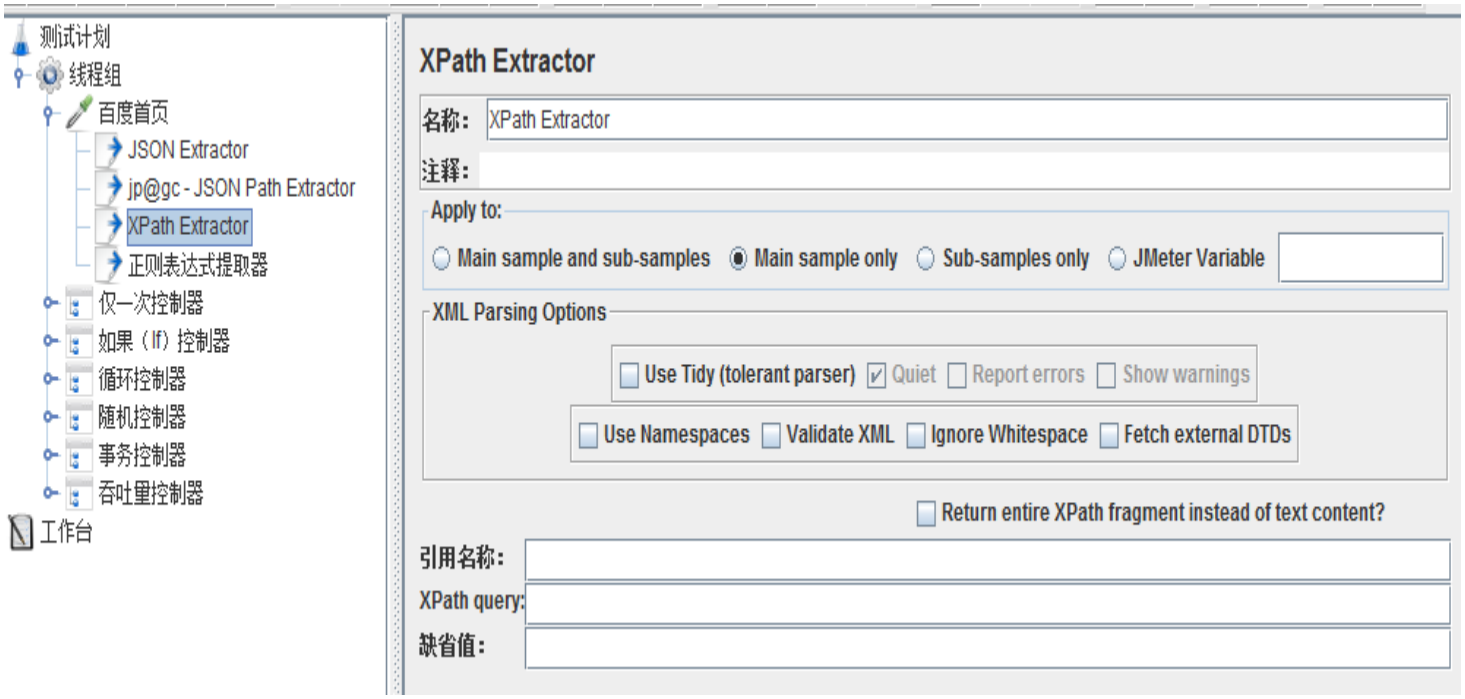
Destination Variable Name： 变量名称，提取到的值将存放在该变量里，后续通过该变量即可引用提取到的数据

JSONPath Expression：JSON 表达式

Default Value：未取到值的时候默认值

具体示例请参见 JSON Extractor 的示例。这里不做详细示例了。

- XPath Extractor



6.png

用于处理响应结果为 xml 格式的内容。

这里对关键参数进行说明：

引用名称：变量名称，提取到的值将存放在该变量里，后续通过该

变量即可引用提取到的数据

XPath query：xpath 表达式

缺省值：未取到值的时候默认值

示例

假如服务端返回如下格式的内容

<title>Apache JMeter</title>

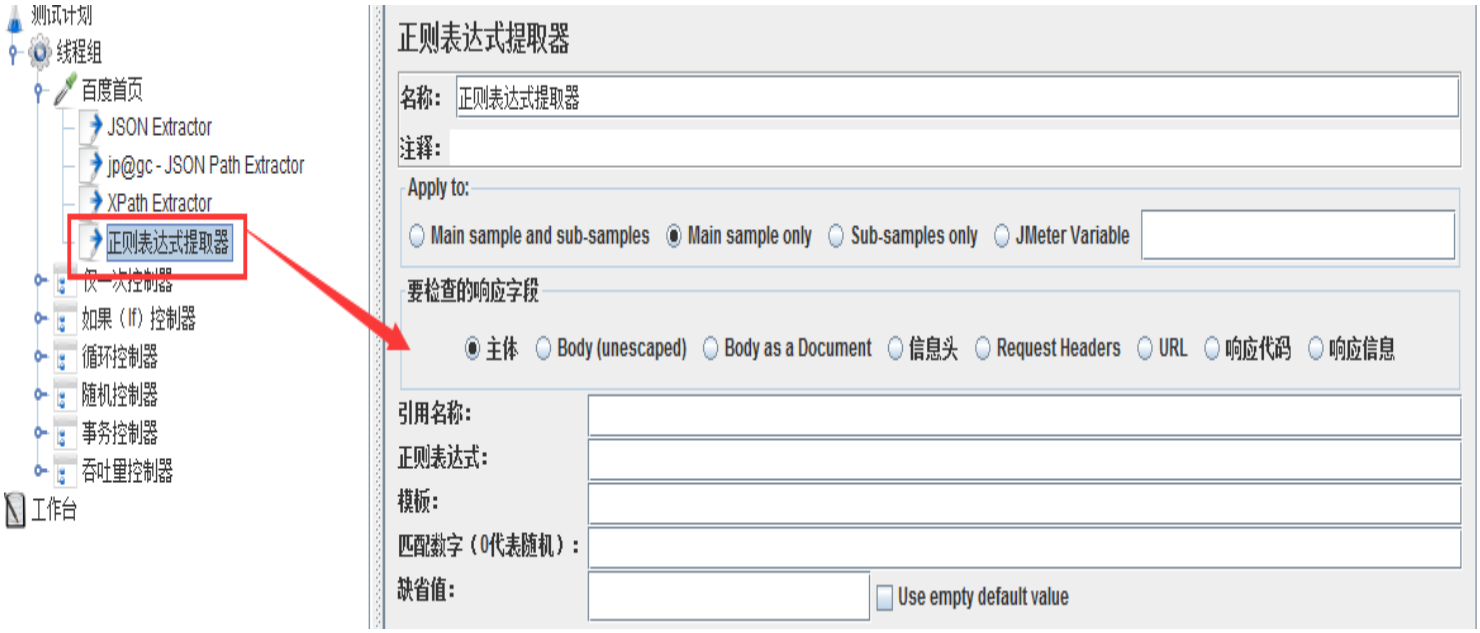
那么我们可以通过，以下 xpath 表达式获取到 Apache JMeter 字符串

//title/text()

将该 xpath 表达式填入在 XPath query 对应输入框中。

- 正则表达式提取器

  这是万能的提取模式了，支持使用正则表达式来提取满足要求的数据。当然你得熟练掌握正则表达式相关知识，才能游刃有余的应用。



7.png

引用名称：变量名称，提取到的值将存放在该变量里，后续通过该变量即可引用提取到的数据

正则表达式：用于匹配目标数据的正则表达式

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

具体的示例这里就不列举了，大家自己去尝试。

9.4 总结

本次主要就后置处理器中常用的 json、xml 及正则表达式处理器进行了分享。在日常测试过程中，这三种后置处理器是必须掌握的，需要深入掌握理解，同时需要对 json、xpath、和正则表达式相关知识有所掌握才行。

第十篇 JMeter 监听器

10.1 前言

在 jmeter 中，通过监听器组件来提供查看、保存、和读取已保存的测试结果功能。

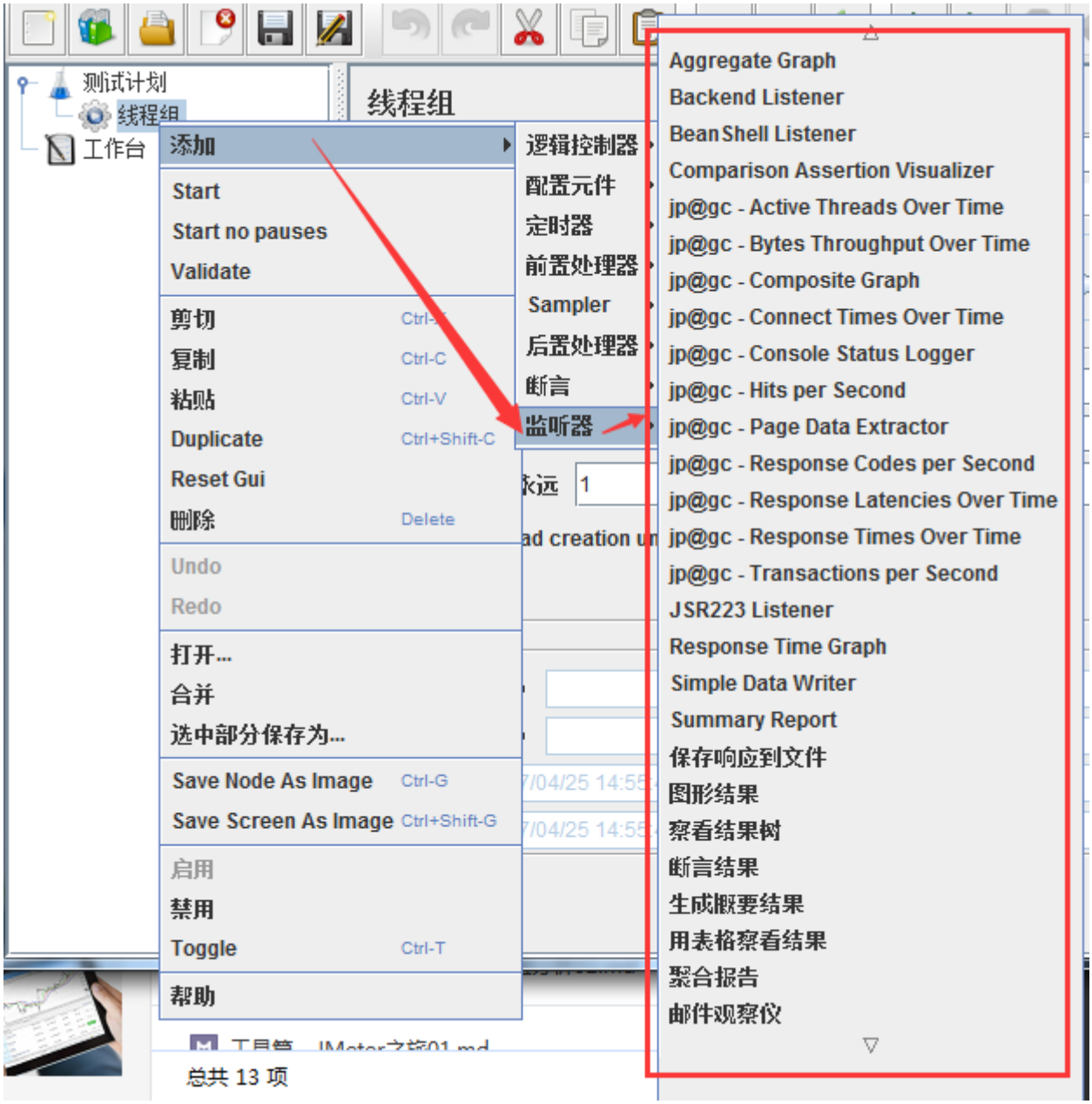
默认情况下，测试结果将被存储为 xml 格式的文件，文件的后缀: ".jtl"。另外一种存储格式为 CSV 文件，该格式的好处就是效率更高，但存储的信息不如 xml 格式详细。

通常情况下，监听器有以下四种类型：

- 树（tree）
- 表（table）
- 图形
- 日志文件

==注：笔者的监听器之所以有这么丰富，是因为安装了更多的插件。==

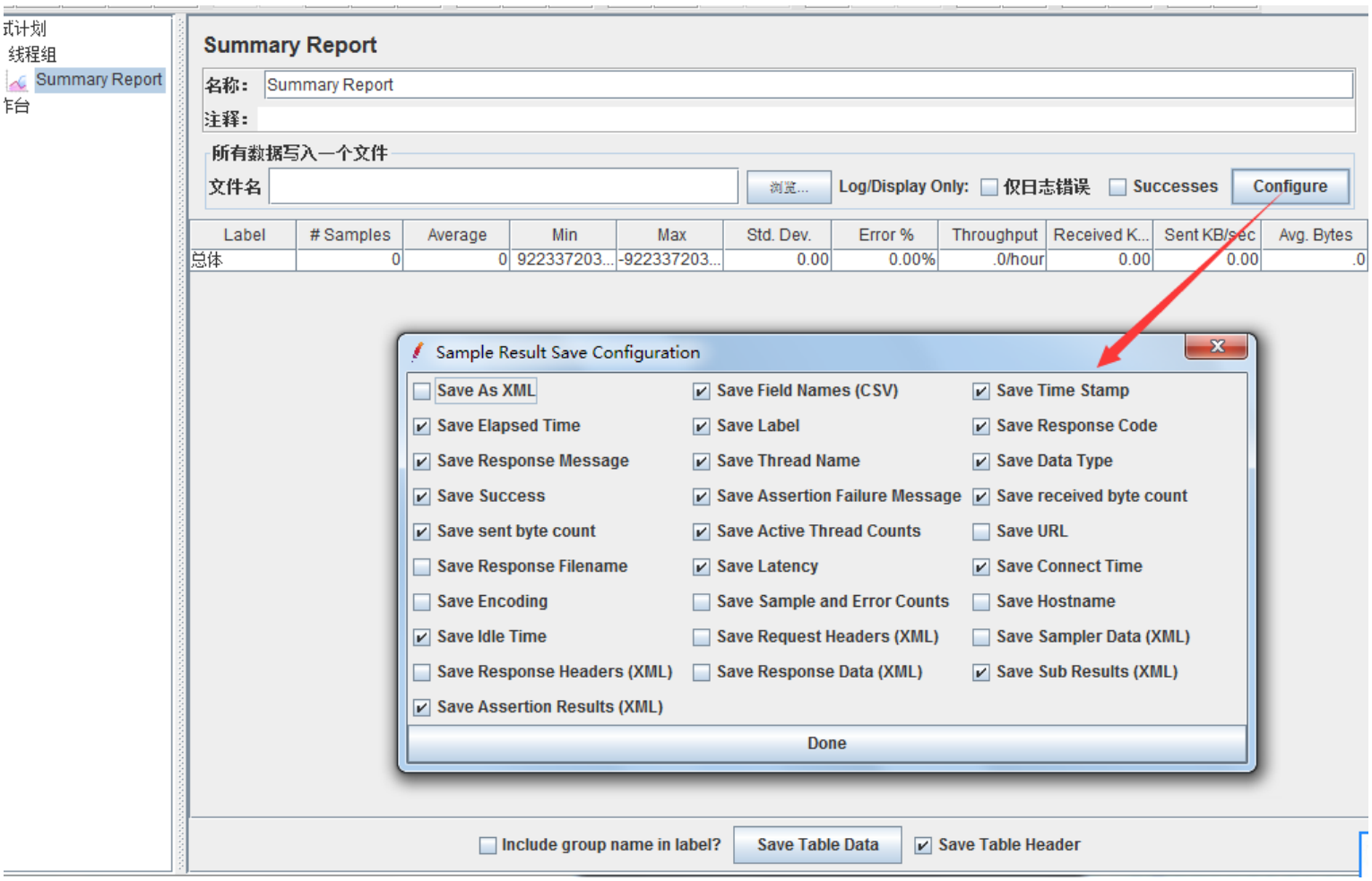
下面我们选取集中常用的监听器进行说明。



1.png

10.2 Summary Report

概要报告，提供了最简要的测试结果信息，同时可以配置将相应的信息保存至指定的文件中（支持 xml、csv 格式的文件）。



2.png

下面我们就每个标签含义进行简单的说明

Label： 请求名称

#Smaples： 请求计数

Average: 请求响应平均耗时

Min： 请求响应最小耗时

Max： 请求响应最大耗时

Std. Dev: 请求响应时间的标准差

Error %： 请求错误率

Throughput： 吞吐量

Received KB/sec: 每秒接收（即响应）的数据量 KB

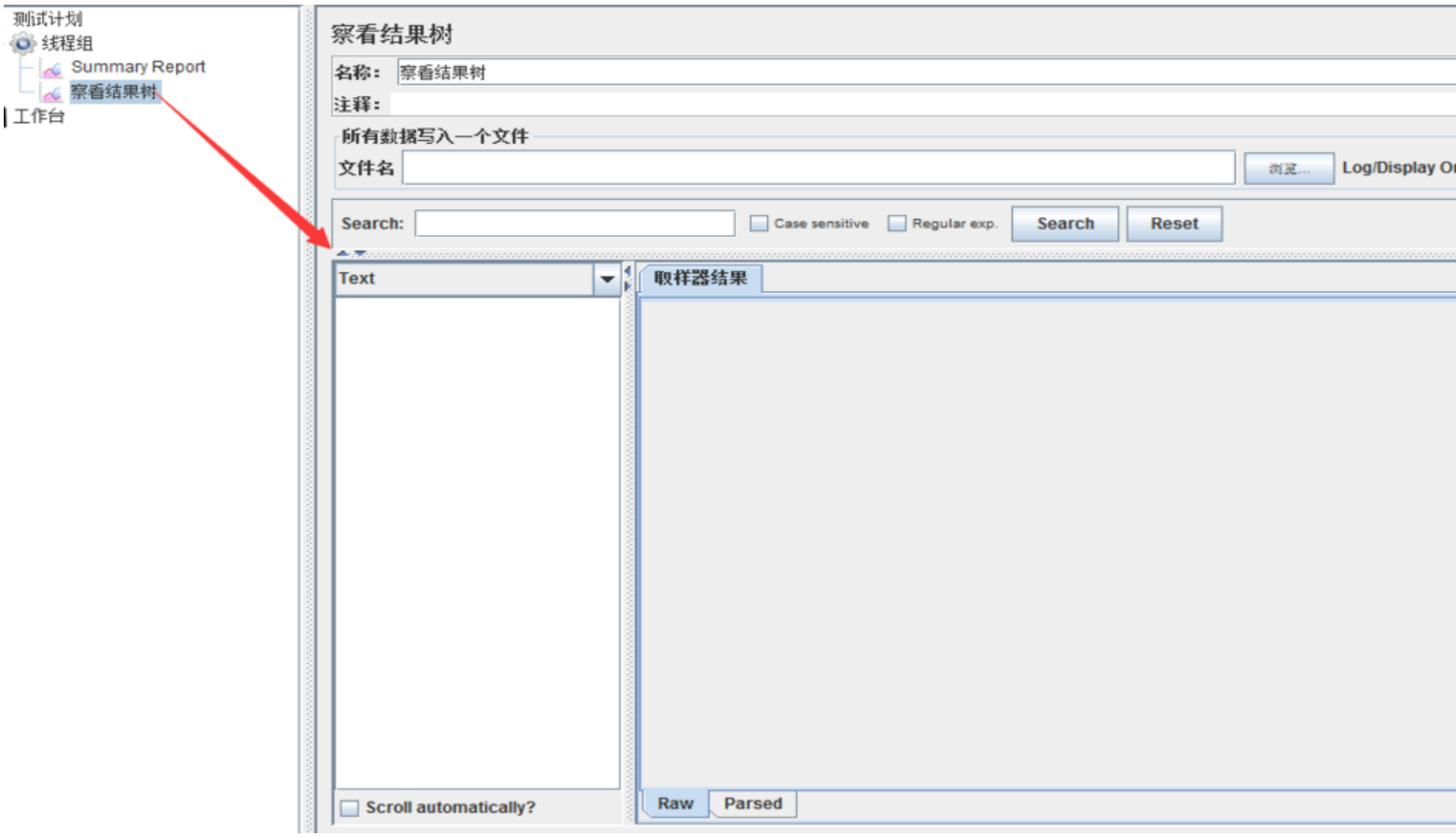
Sent KB/sec: 每秒发送的数据量 KB

Avg. Bytes: 服务端响应的数据的平均值

单击 Configure 按钮，可以配置结果保存各种选项，具体这里不做说明了。

该监听器是笔者在调试 jmeter 项目时常用的监听器之一。

10.3 察看结果树



3.png

该监听器有两个作用

- 查看请求结果，通过的测试通常为绿色。红色则代表失败。
- 查看对应 Sampler 的测试结果的请求、响应数据。

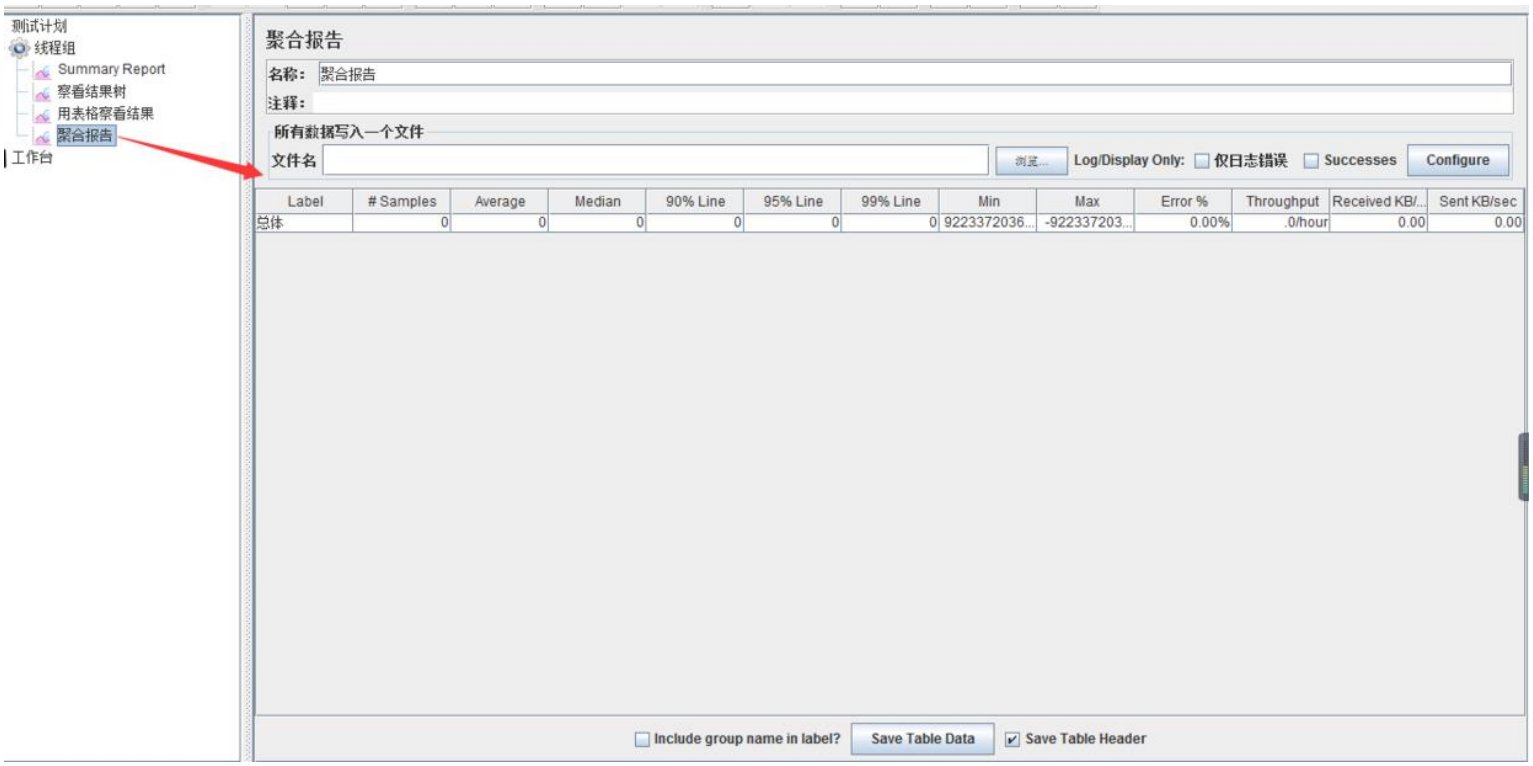
是调试 jmeter 测试的的利器，必须掌握，也是常用的监听器。

不过要注意的是，该监听器笔者推荐做调试用，在实际运行压测时，应该禁用，因为大量请求时，该监听器会造成大 IO 消耗，影响压力机性能。

10.4 聚合报告

聚合报告应该是最详细的报告了，也是最为常用的报告。是大家在压测过程中最常用的监听器。

该监听器对于每个请求，它统计响应信息并提供请求数，平均值，最大，最小值，中位数、90%、95%、错误率，吞吐量(以请求数/秒为单位)和以 kb/秒为单位的吞吐量。



4.png

Label：请求名

#Samples： 请求计数

Average： 请求响应平均耗时

Meian： 中位数，表示 50%的请求在该耗时内完成

90% Line： 表示 90%的请求在该耗时内完成

95% Line： 表示 95%的请求在该耗时内完成

99% Line： 表示 99%的请求在该耗时内完成

Min： 请求响应最小耗时

Max： 请求响应最大耗时

Error %： 请求错误率

Throughput: 吞吐，每秒处理请求数

Received KB/sec: 每秒接收多少 KB 数据

Sent KB/sec: 每秒发送多少 KB 数据

单击 Configure 按钮，可以配置结果保存各种选项，具体这里不做说明了。

10.5 总结

上述三种监听器是笔者日常工作中常用的监听器，对于其他监听器大家可以自行研究。在实际的性能测试过程中，笔者一般使用第三方监控工具或系统。这里就常用的三种进行说明，后续在分享在诊断调优过程和生产运营过程中用到的监控系统和工具。

第十一篇 JMeter 函数和变量

11.1 前言

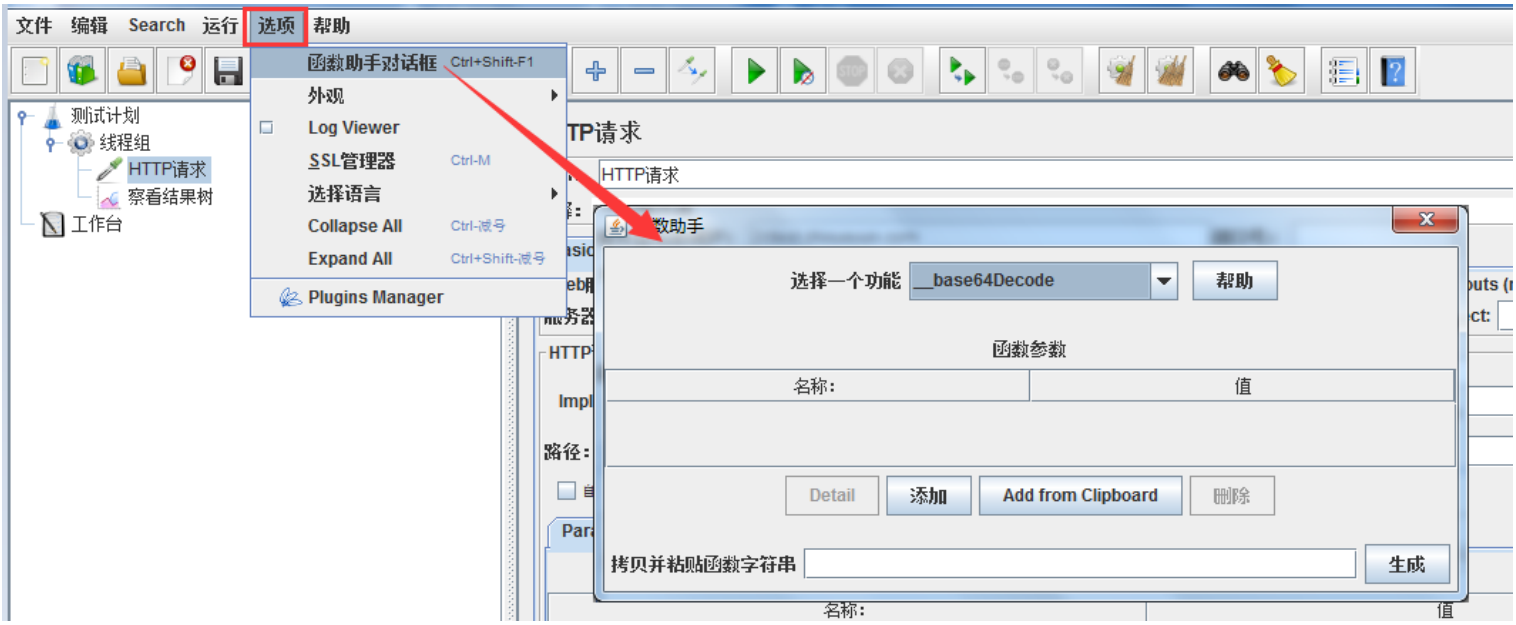
在 jmeter 中提供了功能强大的内置函数来帮助我们处理字符串、文件读写、计算、运行外部脚本等等能力。

要想在项目中切实运用来 jmeter 完成复杂的压测场景，函数和变量是必须掌握的高阶能力。

下面我们就函数和变量进行一一讲解。

11.2 函数

我们在哪可以知道 jmeter 支持哪些函数呢？通过在菜单 “选项” -> "函数助手对话框" 即可打开函数助手。



1.png

通过函数助手，我们可以快速的填充对应的参数来生成我们所需要的函数。

下面我们看一下函数调用示例说明:

${__functionName(param1, param2, param3)}

说明：

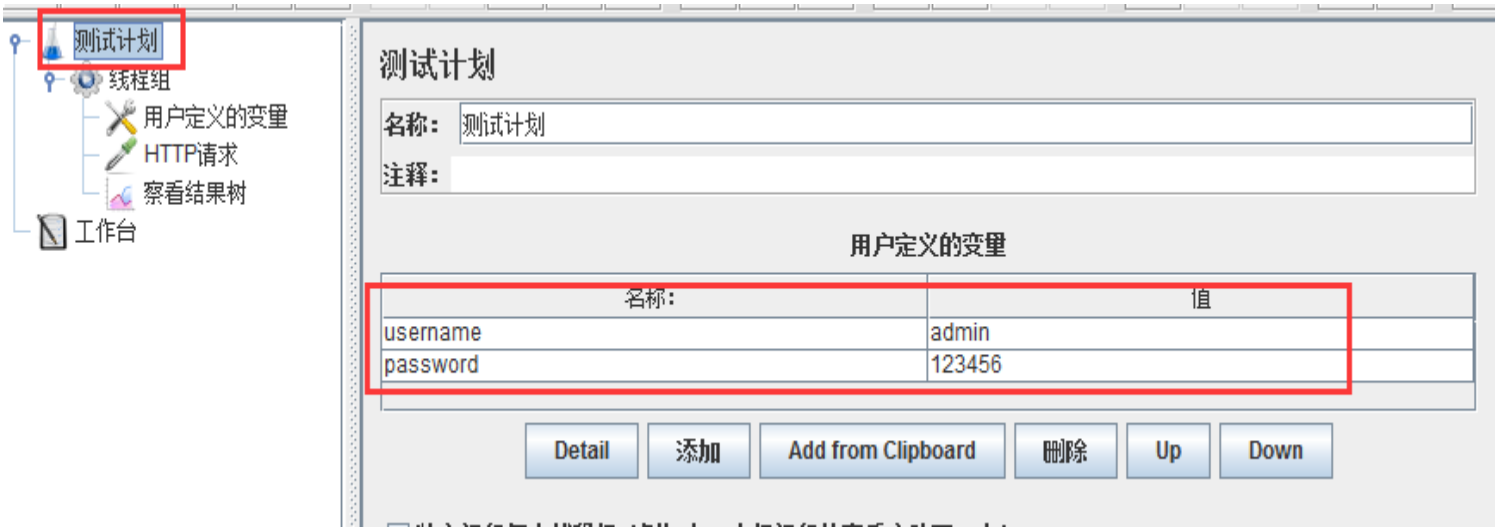
functionName： 指 jmeter 内置函数名称

param1, param2, param3: 指该函数调用时需要传入的参数

11.3 变量

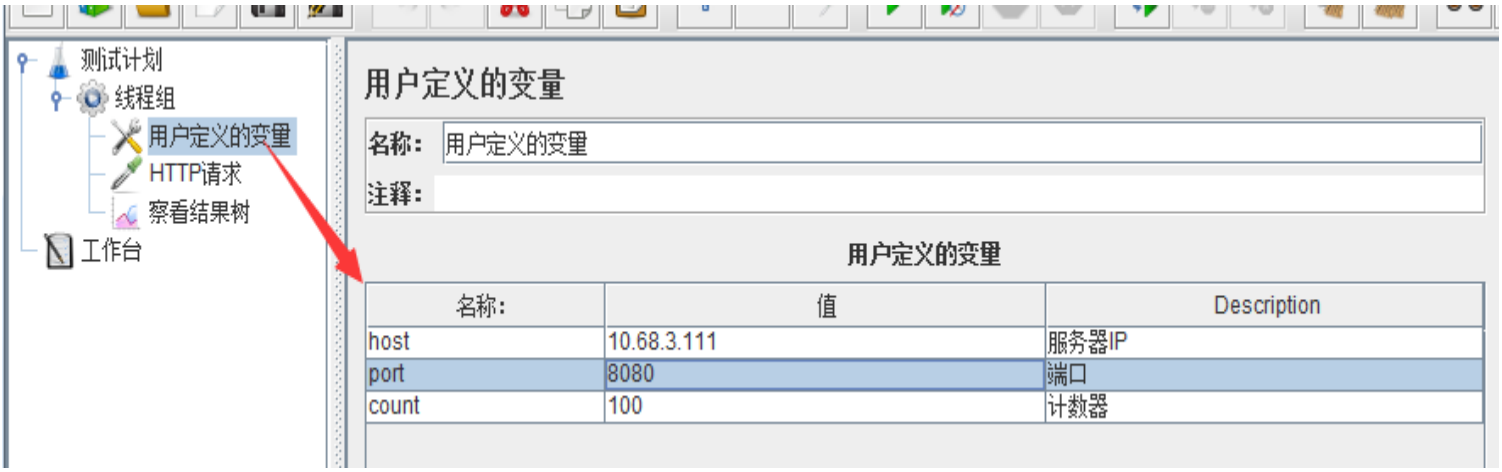
在使用变量前，必须先定义变量，而定义变量有两个地方。

方式一是在测试计划的用户定义的变量处进行定义，如下图



2.png

方式二是“配置元件”中的“用户定义的变量”来进行定义，入下图



3.png

定义了变量，怎么引用呢？ 下面我们展示下引用格式：

${VARIABLE}

VARIABLE: 定义的变量名称

引用前面定义的 username、password 则是

${username}

${password}

同样的道理，引用用户定义的变量组件中定义的 host、port、count 则是

${host}

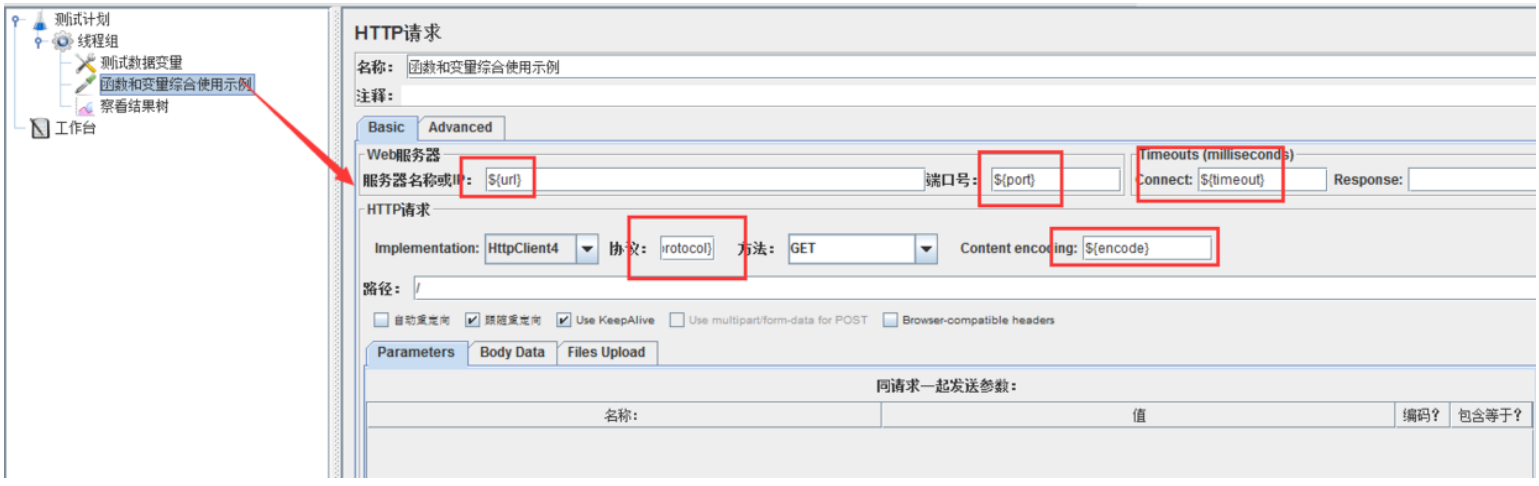
${port}

${count}

下面我们看下如何把函数和变量结合一起应用的简单示例，如下图所示，先定义变量：

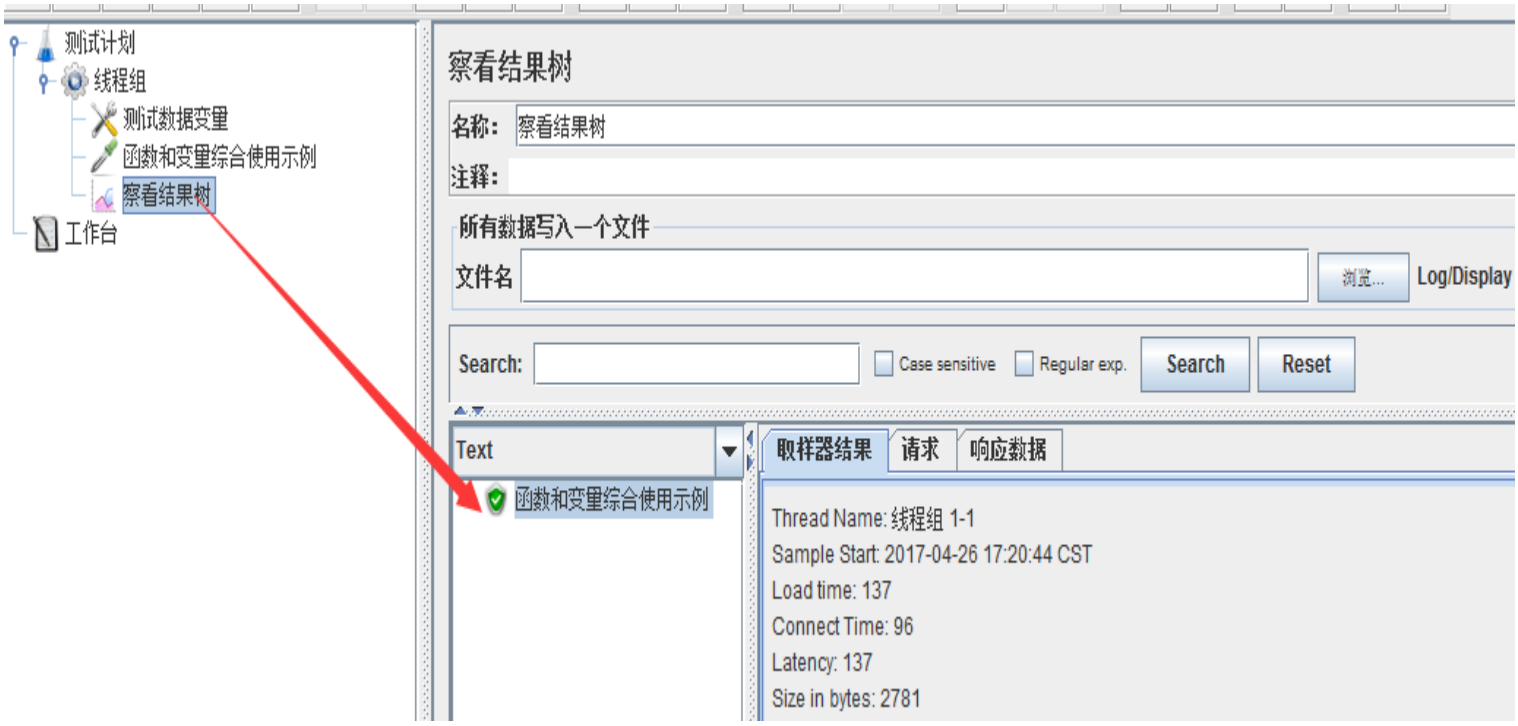
4.png

使用前面定义的变量，来参数化，HTTP 请求相关参数：



5.png

看下请求结果：



6.png

11.4 函数列表

下面我们看下 jmeter 提供的所有内置函数的功能说明及使用示例。

总计七大类型。类型如下：

- 信息类： 用于读取线程、请求名等
- 输入类： 用于读取文件等
- 计算类： 用于计数、求和等
- 脚本类： 用于运行各类脚本，例如 groovy、beanshell 等等
- 属性类： 读取或设置 jmeter 配置
- 变量类： 用于对变量进行操作
- 字符串类： 用于字符串处理

11.4.1 信息类

主要用于获取一些常用的基本信息或是日志输出控制。

序号	函数名	描述
1	threadNum	获取线程数
2	samplerName	获取当前请求的名称（标签）
3	machineIP	获取本机的IP地址
4	machineName	获取本机的名称
5	time	返回指定格式的当前时间
6	log	日志输出，并返回它的输入字符串
7	logn	日志输出，并返回空字符串

7.png

11.4.2 输入类

主要用于从外部文件读取数据，进行参数化或是说关联

序号	函数名	描述
1	StringFromFile	从文件读取一行数据
2	FileToString	读取整个文件数据
3	CSVRead	读取csv格式文件数据
4	XPath	使用xpath读取xml文件中的数据

8.png

11.4.3 计算类

主要用于计算或是随机生成数据

序号	函数名	描述
1	counter	生成一个自增数
2	intSum	整数求和
3	longSum	长整数求和
4	Random	生成一个随机数
5	RandomFromMultipleVars	从一组数据中提取一个数据
6	RandomString	生成随机字符串
7	UUID	生成随机的唯一的UUID

9.png

11.4.4 脚本类

主要用于调用外部脚本或是解析执行脚本

序号	函数名	描述
1	groovy	执行groovy脚本
2	BeanShell	执行BeanShell脚本
3	javaScript	执行javascript脚本
4	jexl2	执行jexl2表达式
5	jexl3	执行jexl3表达式

10.png

11.4.5 属性类

用于读取和设置 jmeter 配置

序号	函数名	描述
1	property	读取属性值
2	P	读取属性值（property简写方法）
3	setProperty	设置属性值

11.png

11.4.6 变量类

主要用于验证变量表达式引用是否正确

序号	函数名	描述
1	split	通过分隔符来拆分传递给它的字符串，并返回原始的字符串。
2	V	执行变量名表达式并返回结果
3	eval	执行字符串表达式，并返回结果
4	evalVar	执行保存在变量中的表达式，并返回结果

12.png

11.6.7 字符串类

用于字符串操作

序号	函数名	描述
1	regexFunction	使用正则表达式来解析之前的响应内容
2	escapeOroRegexpChars	用于转义正则表达式中的字符
3	char	将一串数字转换成unicode字符值，并返回
4	unescape	用于反转义java-escape字符串，并返回
5	unescapeHtml	解码hmlt-encoded字符串
6	escapeHtml	使用html coding编码目标字符串
7	escapeXml	使用xml coding编码目标字符串
8	urldecode	解码Application/x-www-form-urlencoded字符串
9	urlencode	将字符串编码为Application/x-www-form-urlencoded格式
10	TestPlanName	获取当前测试计划名称

13.png

11.5 必须掌握的函数

在上述内容中，并没有把所有的函数都一一列出来，但基本把个大类中主要的函数都已列出，需要大家对其有个基本印象，知道有哪些内置函数，这些函数能解决什么问题，以便在实际项目中走太多弯路。

下面把笔者在实际项目中常用的函数重点列出来。我想这也是大家在项目中常用的，也是重点掌握的，必须熟练能熟练的应用。

注：在本篇中不进行示例讲解，后续实战专题在分享。

从文件读取数据，进行参数化

- StringFromFile
- CSVRead

- XPath

脚本支持

- BeanShell（推荐这个）
- groovy

随机数据生成

- RandomString
- UUID

字符串处理

- urldecode
- urlencode
- char

注：并不是其他函数不重要，而是上述函数是笔者日常项目实践中用得最为频繁，建议必须掌握的。

11.6 总结

本文就 jmeter 函数和变量进行了分享，这是进一步掌握 jmeter 必备的技能。也是在项目实践中进行参数化、关联必备的技能。对于所有函数要做到心中有数，对于关键重点的函数要做到随时会用，灵活应用