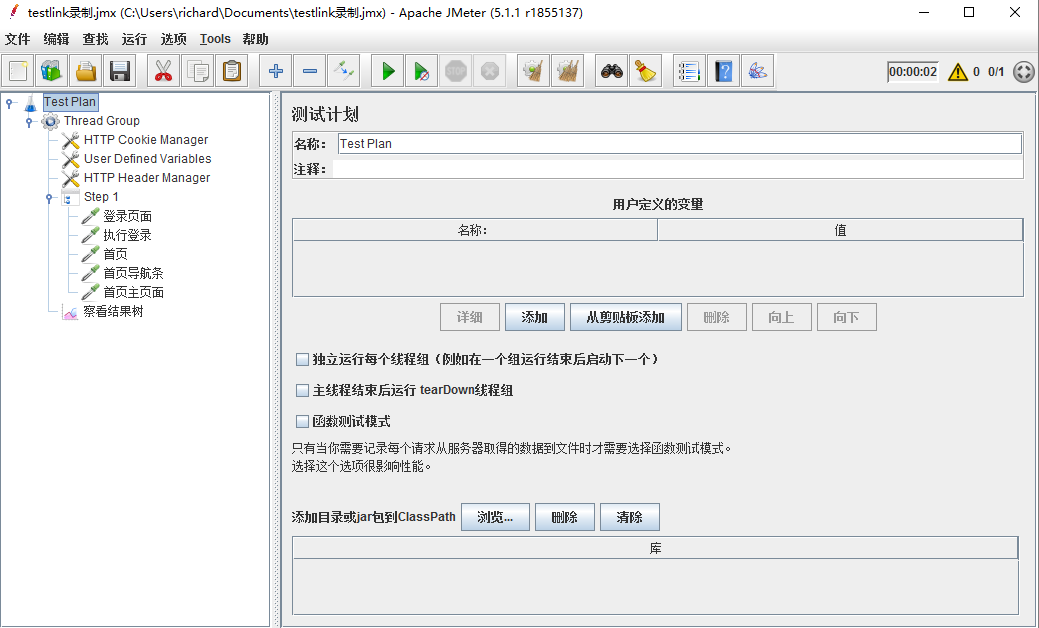
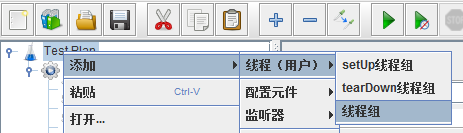
**Jmeter常用组件**

## 测试计划(Test Plan):

用来描述一个性能测试，包含与本次性能测试所有相关的功能。也就说本的性能测试的所有内容是于基于一个计划的。测试计划页面可以添加一些与用户自定义的变量和第三方jar包等，如数据库驱动jar包

****

## 线程组（Thread Group）：

****

有三个添加线程组的选项，名字不一样， 创建之后，其界面是完全一样的。之前的版本只有一个线程组的名字。现在多一个setUp theread Group 与terDown Thread Group

**1) setUp线程组（setup thread group）**

　　一种特殊类型的ThreadGroup的，可用于执行预测试操作。这些线程的行为完全像一个正常的线程组元件。不同的是，这些类型的线程执行测试前进行定期线程组的执行。

**2) teardown 线程组（teardown thread group）**.

　　一种特殊类型的ThreadGroup的，可用于执行测试后动作。这些线程的行为完全像一个正常的线程组元件。不同的是，这些类型的线程执行测试结束后执行定期的线程组。

　　可能你还是不太理解他们与普通的线程组有什么不同。 如果您用过junit，想必你不会对setup ，teardown这2个字眼陌生。 即时没用过，也没关系。 熟悉loadrunner的应该知道，loadrunner的脚本除了action里是真正的脚本核心内容，还有初始化“环境”的初始化脚本和测试完毕后对应的清除信息的脚本块。 那么这里 setup thread group 和 teardown thread group 就是分别指这两部分。  其实从本质上来看，他们并没有什么不同。

**3) thread group(线程组).**

     这个就是我们通常添加运行的线程。通俗的讲一个线程组,，可以看做一个虚拟用户组，线程组中的每个线程都可以理解为一个虚拟用户。线程组中包含的线程数量在测试执行过程中是不会发生改变的。

Ramp-up Period(in Seconds)：表示每个用户启动的延迟时间，上述设置为0秒，则表示立即启动所有用户，如果设置为100秒，那么系统将会在100秒结束前启动这100个用户，开始用户的延迟为1秒。  
  
循环次数：如果你要限定循环次数为5次的话，可以取消永远的那个勾，然后在后面的文本框里面填写5；勾上永远，表示如果不停止或者限定时间将会一直执行下去， 是为了方便调度器的调用。  
  
启动时间：表示脚本开始启动的时间，当不想立即启动脚本，但是启动脚本的时间不再电脑旁的时候，你可以设定一个启动的时间，然后再运行那里点击启动，系统将不会立即运行，而是会等到你填写的时间才开始运行。  
  
结束时间：与启动时间对应，表示脚本结束运行的时间  
  
持续时间：表示脚本持续运行的时间，以秒为单位，比如如果你要让用户持续不断登录1个小时，你可以在文本框中填写3600。如果在1小时以内，结束时间已经到达，它将会覆盖结束时间，继续执行。

启动延迟：表示脚本延迟启动的时间，在点击启动后，如果启动时间已经到达，但是还没有到启动延迟的时间，那么，启动延迟将会覆盖启动时间，等到启动延迟的时间到达后，再运行系统。

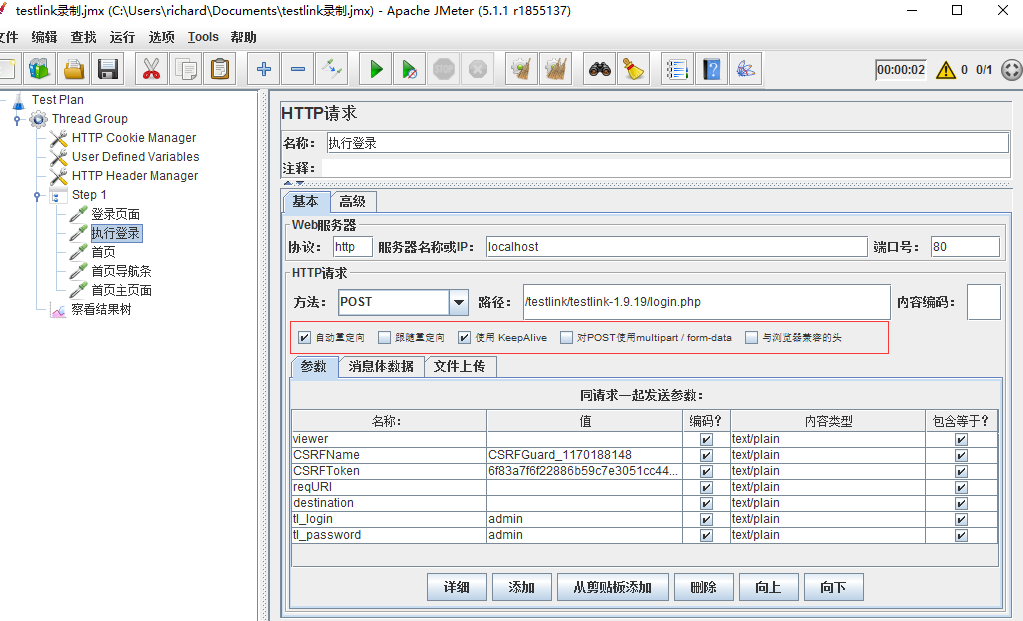
****

## ****取样器（Sampler）:****

取样器（Sample）是性能测试中向服务器发送请求，记录响应信息，记录响应时间的最小单元，JMeter 原生支持多种不同的sampler ，如 HTTP Request Sampler 、 FTP  Request Sample 、TCP  Request Sample 、JDBC Request Sampler 等，每一种不同类型的 sampler 可以根据设置的参数向服务器发出不同类型的请求。

****

## http请求（http requst）

****

**自动重定向：**

当发送HTTP请求后若响应为301/302，Jmeter会自动重定向到对应的新页面，但不会记录重定向的请求和响应内容，只有GET和HEAD请求才能使用自动重定向。勾选此项则“跟随重定向”失效；自动重定向可以自动转向到最终目标页面，但是JMeter是不记录重定向过程内容的，比如在查看结果树中是无法找到重定向过程内容的（A重定向到B，此时只记录B的内容不记录A的内容，A的响应内容我们暂时且叫过程内容），如果此时你想做关联，那就比较遗憾了，你无法关联到。

**跟随重定向：**

Http Request取样器的默认选项，但响应Code是3XX时（比如301是重定向），自动跳转至目标地址。与自动重定向不同，JMeter会记录重定向过程中的所有请求响应，在查看结果树时可以看到服务器返回的内容，所以你可以对响应的内容做关联   
比如你要测试登录，你把POST请求改为跟随重定向就可以了

**Use keep Alive** ：

当该选项被选中时，jmeter 和目标服务器之间使用 Keep-Alive方式进行HTTP通信，默认选中。在默认HTTP实现下它不起作用，因为连接重用不在用户控制之下。但在Apache HttpComponents HttpClient下它是起作用的。

**Use multipart/from-data for HTTP POST** ：

使用 multipart/from-data或application/x-www-form-urlencoded方法发送HTTP POST 请求，默认不选中。

**Browser-compatible headers**：

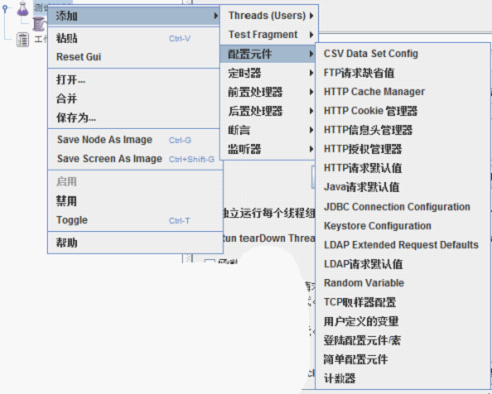
浏览器header兼容设置。当使用了 multipart/from-data，它会优先于header中的Content-Type and Content-Transfer-Encoding设置，只发送header的Content-Disposition。

## 逻辑控制器（Logic Controller）

****

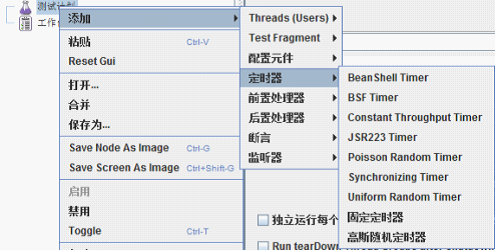
　　逻辑控制器，包括两类元件，一类是用于控制test plan 中 sampler 节点发送请求的逻辑顺序的控制器，常用的有 如果（If）控制器 、switch Controller 、Runtime Controller、循环控制器等。另一类是用来组织可控制 sampler 来节点的，如 事务控制器、吞吐量控制器。

## 配置元件（Config Element）

****

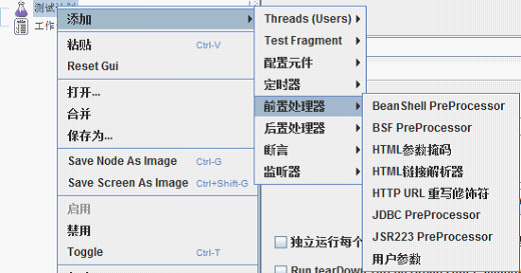
　　配置元件（config element）用于提供对静态数据配置的支持。CSV Data Set config 可以将本地数据文件形成数据池（Data Pool），而对应于HTTP Request Sampler和 TCP Request Sampler等类型的配制无件则可以修改Sampler的默认数据。

## 定时器（Timer）

****

 　　定时器（Timer）用于操作之间设置等待时间，等待时间是性能测试中常用的控制客户端QPS的手端。类似于LoadRunner里面的“思考时间”。JMeter 定义了Bean Shell Timer、Constant Throughput Timer、固定定时器等不同类型的Timer。

## 前置处理器（Per Processors）

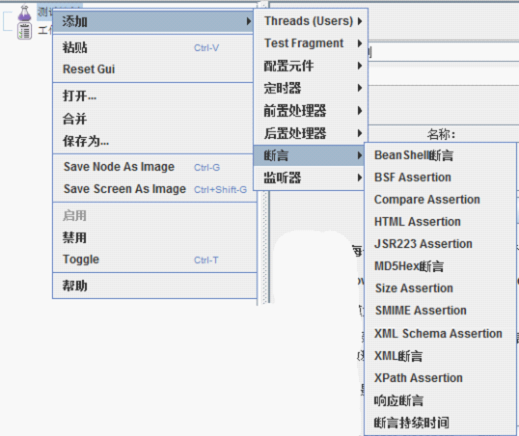
****

　   用于在实际的请求发出之前对即将发出的请求进行特殊处理。例如，HTTP URL重写修复符则可以实现URL重写，当RUL中有sessionID 一类的session信息时，可以通过该处理器填充发出请求的实际的sessionID 。

## 后置处理器（Post Processors）

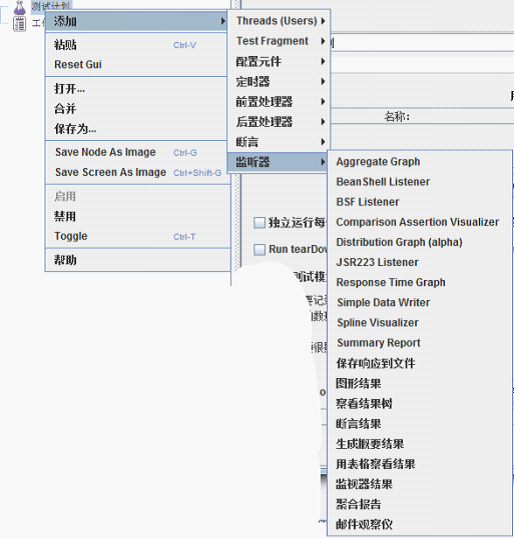
　　用于对Sampler 发出请求后得到的服务器响应进行处理。一般用来提取响应中的特定数据（类似LoadRunner测试工具中的关联概念）。例如，XPath  Extractor 则可以用于提取响应数据中通过给定XPath 值获得的数据。

## 断言（Assertions）

****

　　 断言用于检查测试中得到的相应数据等是否符合预期，断言一般用来设置检查点，用以保证性能测试过程中的数据交互是否与预期一致。

## 监听器（Listener）

****

 　　这个监听器可不是用来监听系统资源的元件。它是用来对测试结果数据进行处理和可视化展示的一系列元件。 图行结果、查看结果树、聚合报告。都是我们经常用到的元件。

到此，我们已经简单了解了JMeter的基本组成元件，后续我们通过设置参数和聚合报告来执行一个简单的性能测试