**Jmeter之逻辑控制器(Logic Controller)**

**前言：**

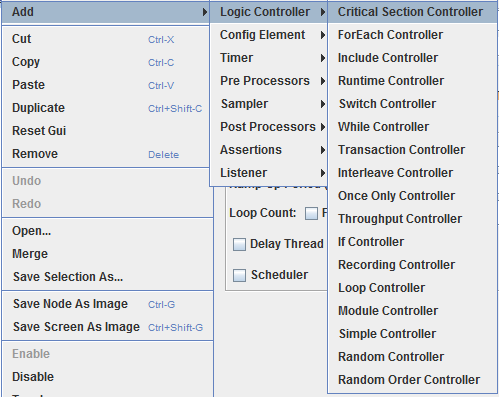
1、 Jmeter官网对逻辑控制器的解释是：“Logic Controllers determine the order in which Samplers are processed.”。意思是说，逻辑控制器可以控制采样器(samplers)的执行顺序。由此可知，控制器需要和采样器一起使用，否则控制器就没有什么意义了。放在控制器下面的所有的采样器都会当做一个整体，执行时也会一起被执行。

2、JMeter中的Logic Controller分为两类：

　　a) 控制测试计划执行过程中节点的逻辑执行顺序，如：Loop Controller、If Controller等；

　　b) 对测试计划中的脚本进行分组、方便JMeter统计执行结果以及进行脚本的运行时控制等，如：Throughput Controller、Transaction Controller。

3、Jmeter提供如下这么多的控制器：



**一、简单控制器（Simple Controller）：**

　　作用：这是Jmeter里最简单的一个控制器，它可以让我们组织我们的采样器和其它的逻辑控制器（分组功能），提供一个块的结构和控制，并不具有任何的逻辑控制或运行时的功能。

**二、循环控制器（Loop Controller）：**

　　作用：指定其子节点运行的次数，可以使用具体的数值（如下图，设置为5次），也可以使用变量

　　1、Forever选项：勾选上这一项表示一直循环下去

　　2、如果同时设置了线程组的循环次数和循环控制器的循环次数，那循环控制器的子节点运行的次数为两个数值相乘的结果。

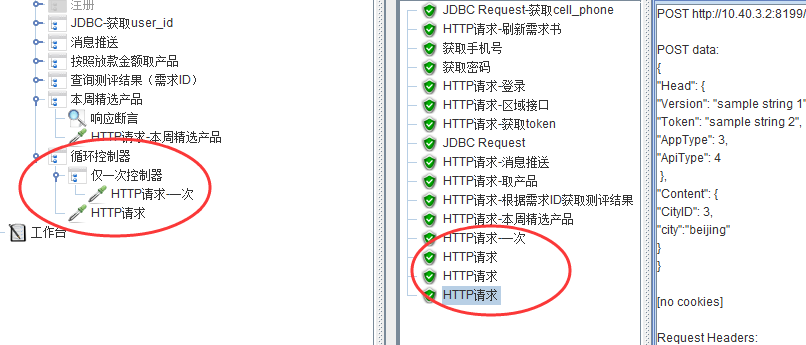
 如上设置了5次，则运行时，该循环控制器下的sample会执行5次。

**三、仅一次控制器（Once Only Controller）：**

　　作用：在测试计划执行期间，该控制器下的子结点对每个线程只执行一次，登录场景经常会使用到这个控制器。

注意：将Once Only Controller作为Loop Controller的子节点，Once Only Controller在每次循环的第一次迭代时均会被执行。

如下例：Once Only Controller作为Loop Controller的子节点只会执行一次，Loop Controller下的其他sample执行了三次（循环次数）

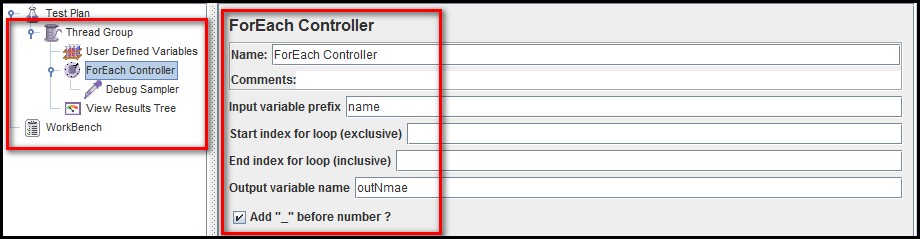


**四、ForEach控制器（ForEach Controller）：**

 　　作用：ForEach控制器一般和用户自定义变量一起使用，其在用户自定义变量中读取一系列相关的变量。该控制器下的采样器或控制器都会被执行一次或多次，每次读取不同的变量值。如下图：

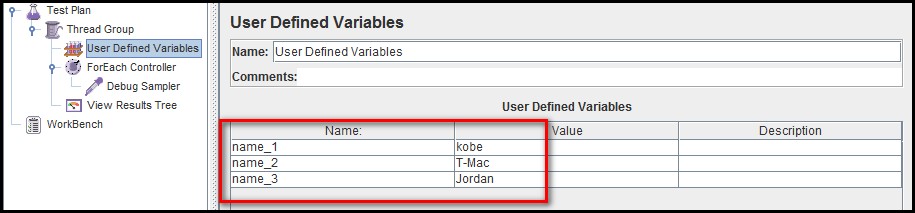
**参数:**

* Input Variable Prefix：输入变量前缀
* Output variable name：输出变量名称
* Start index for loop(exclusive)：循环开始的索引（这里如果不填写，默认从1开始，如果没有1开始的变量，执行时会报错）
* End index for loop(inclusive)：循环结束的索引
* Add”\_”before number：输入变量名称中是否使用“\_”进行间隔。



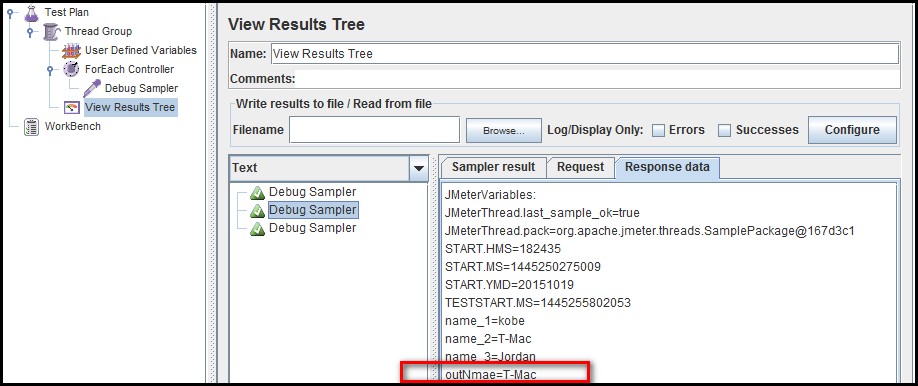
用户自定义变量：

　　变量名前缀为ForEach Controller中Input variable prefix定义的name + 下划线（上图中我们勾选了下划线）+数字编号



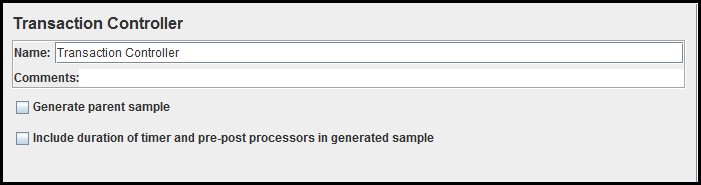
执行结果：

　　总共执行了3次，每次执行时会把获取到的变量值赋值给输出变量outNmae，其它地方可以通过${outNmae}进行调用。



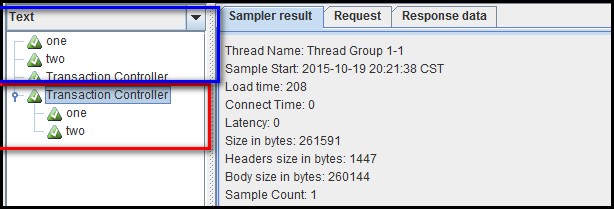
**五、事务控制器（Transaction Controller）：**

　　作用： 事务控制器会生产一个额外的采样器，用来统计该控制器子结点的所有时间。

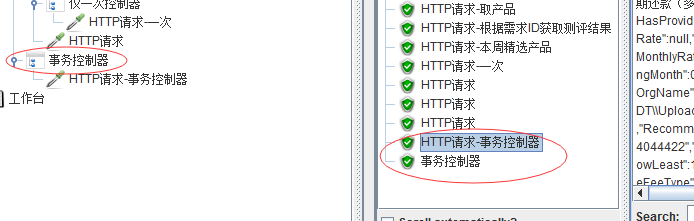


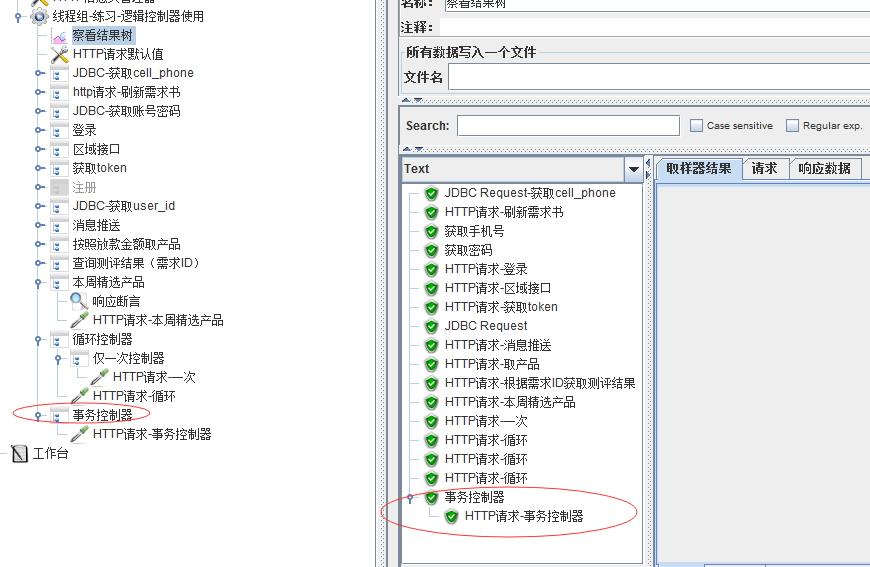
**参数：**

* Generate parent sample：(选中这个参数结果展示如下图红框，否则显示为下图蓝框)
* Include duration of timer and pre-post processors in generated sample：选中这一项会统计定时器(timer)的时间，否则只统计采样器(sample)的时间



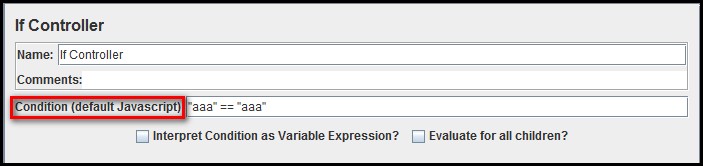
如下例（不勾选/勾选Generate parent sample）：





**六、If 控制器（If Controller）：**

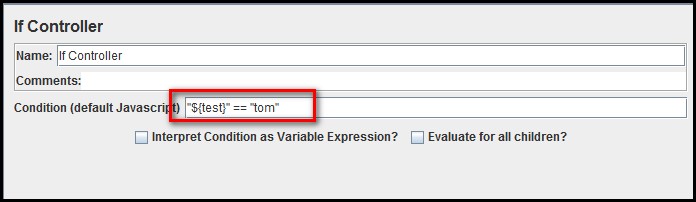
 　　作用：根据给定表达式的值决定是否执行该节点下的子节点，默认使用javascript的语法进行判断(如下图红框内的文字)。



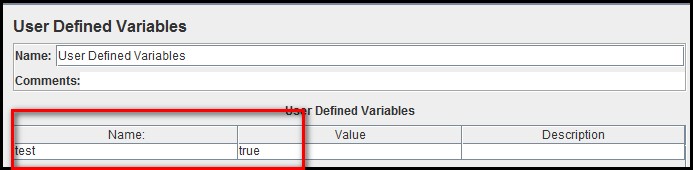
**参数：**

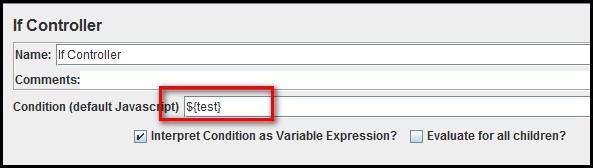
* Interpret Condition as Variable Expression?：选中这一项时表示：判断变量值是否等于字符串true（不区分大小写）
* Evaluate for all children：如果选中这一项，在每个子结点执行前都会计算表达式

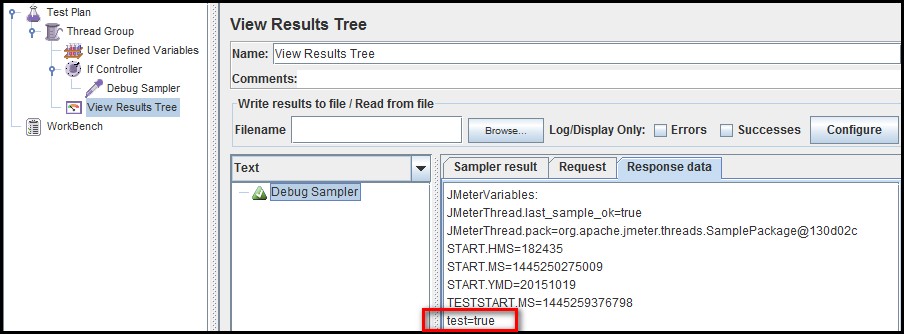
示例一：使用变量的方式进行判断：



示例二：选中Interpret Condition as Variable Expression？







**七、Switch控制器（Switch Controller）：**

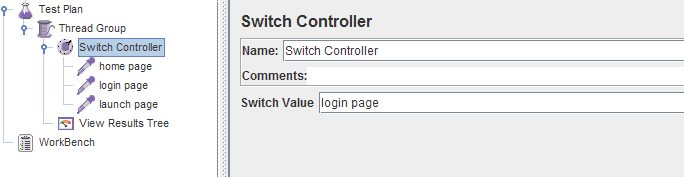
　　作用：Switch控制器通过给该控制器中的Value赋值，来指定运行哪个采样器。有两种赋值方式：

* 第一种是数值，Switch控制器下的子节点从0开始计数，通过指定子节点所在的数值来确定执行哪个元素。
* 第二种是直接指定子元素的名称，比如采样器的Name来进行匹配。当指定的名称不存在时，不执行任何元素。

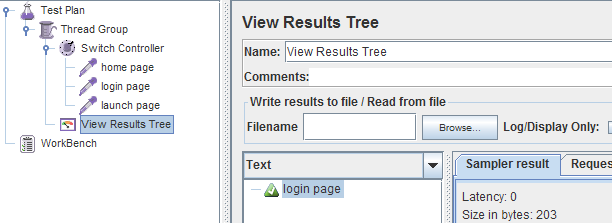
当Value为空时，默认执行第1个子节点元素。

示例：

1、Switch Controller选择的值为login page



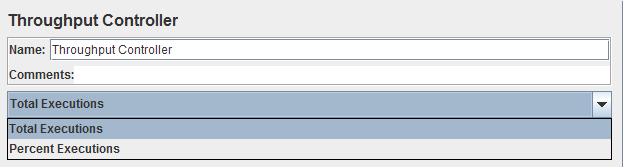
2、执行结果：



**八、吞吐量控制器(Throughput Controller):**

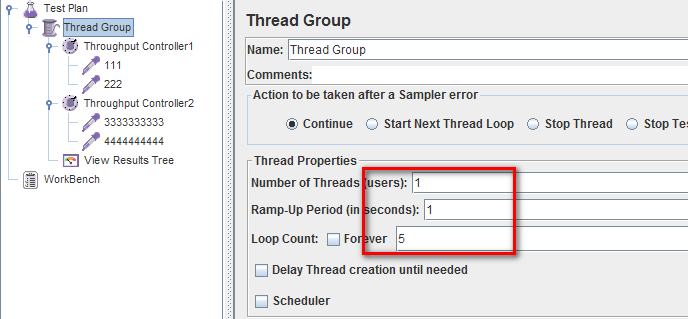
 　　作用：控制其下的子节点的执行次数与负载比例分配，也有两种方式：

* Total Executions：设置运行次数
* Percent Executions：设置运行比例(1~100之间)

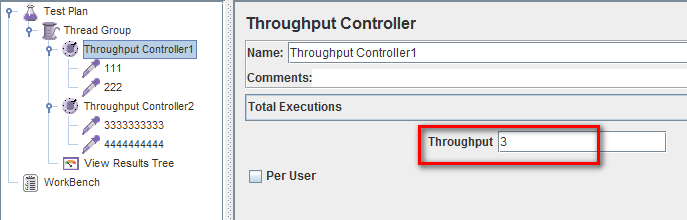
 根据线程组设置页面的循环次数来计算

示例：

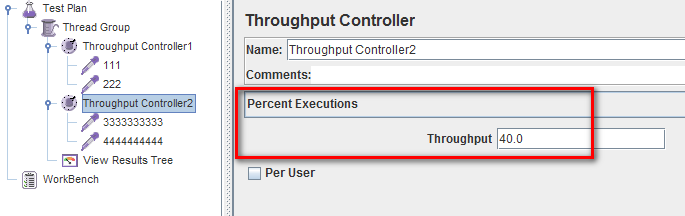
1、设置线程组循环5次：



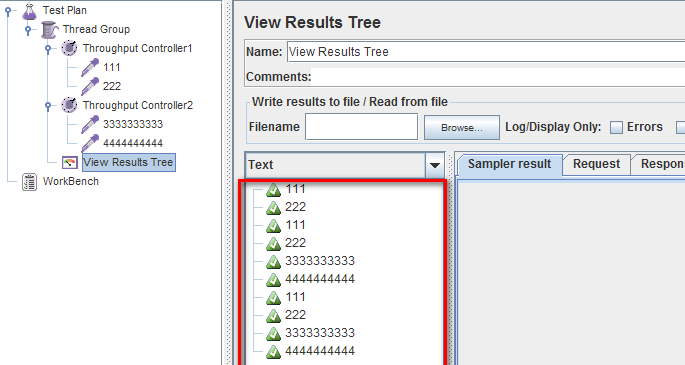
2、Throughput Controller1的子结点执行3次：



3、Throughput Controller2的子结点执行（40% \* 线程组循环次数5）= 2次：



执行结果：



**九、随机控制器(Random Controller):**

 　　作用：随机执行其下的所某个子结点

**十、随机顺序控制器(Random Order Controller):**

 　　作用：随机执行其下的所有子结点