**性能测试-逻辑控制器&定时器**

**回顾：**

（1）JMeter断言

**响应断言：**

 **作用：在任意响应格式中，都可以使用**

 **配置：**

○ **实际要校验的字段**

○ **校验规则**

○ **预期结果**

**JSON断言：**

 **作用：响应格式为JSON数据时，使用JSON断言**

 **配置：**

 **JSON路径**

○ **预期结果**

持续时间断言：

 作用：检查响应时间是否超出预期时间

 配置：预期响应时间

（2）JMeter关联

概念：请求依赖关系， 一个请求的响应作为另一个请求的参数进行传递

**正则表达式提取器：**

 **作用：可以针对任意格式的响应数据进行提取**

 **配置：**

○ **变量名**

 **正则表达式：左边界（.\*?）右边界**

 **模板： $n$ 。提取第几个括号的（提取第几组值）**

○ **匹配数字： 1表示第一个值， -1表示全部值， 0表示随机**

XPath提取器：

 作用：可以针对HTML格式的响应数据进行提取

 配置：

 变量名

 xpath表达式。//标签， //标签[@属性名=属性值]

○ 匹配数字： 1表示第一个值， -1表示全部值， 0表示随机

**JSON提取器：**

 **作用：可以针对JSON格式的响应数据进行提取**

 **配置：**

○ **变量名**

○ **JSON路径。格式： $.键名.键名**

JMeter属性：

 作用：跨线程组关联，跨线程组实现数据的传递

 函数及调用：

○ 保存JMeter属性： setProperty —— 放入到BeanShell取样器中执行

○ 读取JMeter属性： Property —— 放入到第二个线程组的HTTP请求中

 步骤：

 添加第一个线程组

 添加HTTP请求

○ 添加JSON/Xpath/正则表达式提取器 —— 提取数据

○ 添加BeanShell取样器 —— 执行setProperty函数

○ 添加第二个线程组

 添加HTTP请求 —— Property函数读取JMeter属性

○ 查看结果树

（3）JMeter脚本录制

原理： JMeter作为代理服务器，所有请求和响应数据都会经过JMeter，Jmeter再还原为脚本

步骤：

 添加代理服务器，并配置

○ 端口

○ 目标控制器

○ 过滤条件

 在PC上设置浏览器代理，指向JMeter代理服务器

 启动JMeter代理服务器

 在浏览器上操作并还原出脚本

**目标：**

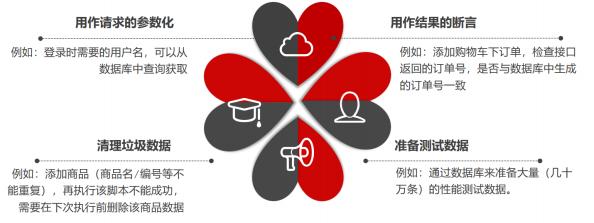
 掌握Jmeter直连数据库

 理解Jmeter逻辑控制器

 理解Jmeter常用定时器

**性能测试工具**

**直连数据库的作用：**



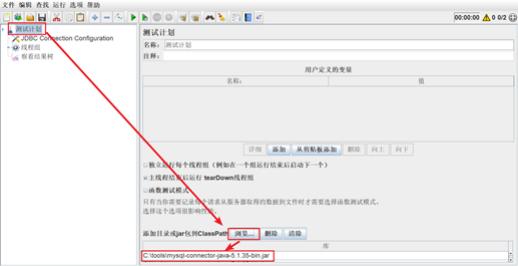
**Jmeter直连数据库：**

**步骤：**

•添加MySQL驱动jar包

-方式一：在测试计划面板点击“浏览 …“按钮，将你的JDBC驱动添加进来

-方式二：将MySQL驱动jar包放入到lib/ext目录下，重启JMeter



•配置数据库连接信息

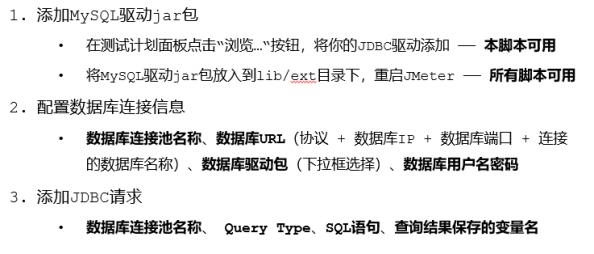
-添加方式：测试计划 --> 线程组--> (右键添加) 配置元件 --> JDBC Connection Configuration





**案例1：**

连接tpshop商城数据库获取商品名包含：小米手机5的商品id



**需求：**

1、在tpshop首页中搜索商品名称“长虹(CHANGHONG) 49A1U 49英寸双64位4K超清智能网络LED液晶 电视”，

请求URL：<http://127.0.0.1/Home/Goods/search.html>

请求方法： GET

请求参数： q=

2、在搜索的响应数据中检查是否有该商品对应的商品链接： /Home/Goods/goodsInfo/id/65.html

（id ：65需要通过数据库查询返回，才保证每次都是有效的）

**编写JDBC脚本步骤（搜索指定商品，在返回结果中检查是否包含指定商品的ID的详情URL）：**

1.添加线程组

2.添加 JDBC Connection Configuration

3.添加 JDBC request

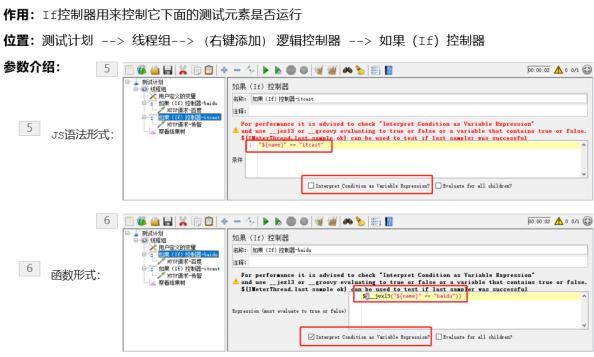
4.添加 HTTP请求 - 搜索商品

5.在搜索HTTP请求下，添加**响应断言**，检查响应结果中的字符 是否**包含** \*\*商品URL。引用格式： ${变 量名\_索引}

6.添加查看结果树

**逻辑控制器：**

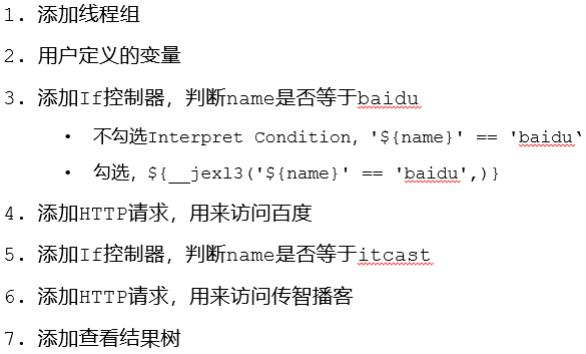
**如果（if）控制器：**



**案例：**

1、使用‘用户定义的变量’定义一个变量name ， name的值可以是baidu或itcast

2、根据name的变量值实现对应网站的访问

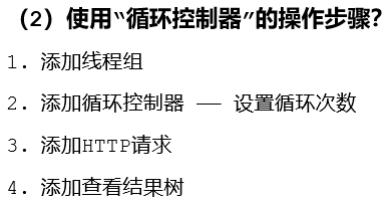


**循环控制器：**



**案例：**

1、循环访问百度10次



（3）思考：线程组属性可以控制循环次数，那么循环控制器有什么用？

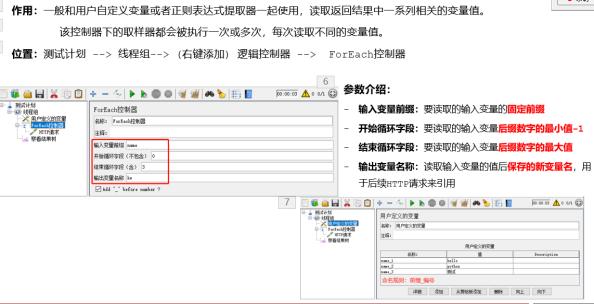
不同点：线程组的循环次数对线程组下的所有HTTP请求有效，循环控制器的循环次数对子节点下的 HTTP请求有效

关联：循环控制器次数为M，线程组循环次数为N，

 循环控制器下的请求运行： M \* N 次

 线程组下，非循环控制器下的请求运行： N次

**ForEach控制器：**



**与用户定义的变量配合使用（案例1）：**

|  |
| --- |
| 案例1需求  1. 有一组关键字 [hello,python,测试]，使用用户定义的变量存储  2. 要依次取出关键字，并在百度搜索，例如：https://www.baidu.com/s?wd=hel |

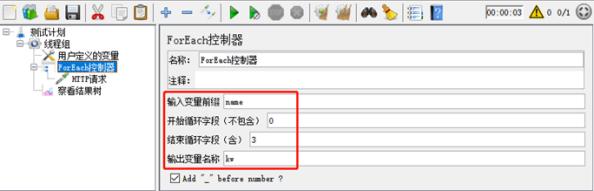
步骤：

1.添加线程组

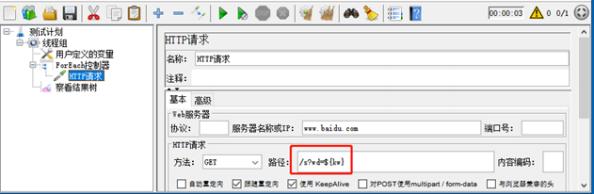
2.添加用户定义的变量



3.添加ForEach控制器



4.添加HTTP请求-百度



5.添加查看结果树

**与正则表达式配合使用（案例2）：**

|  |
| --- |
| 案例2：  1. 访问传智播客首页<http://www.itcast.cn>，获取首页中的地址信息，并全部保存下来 2. 要依次取出地址关键字，并在百度搜索，例如：https://www.baidu.com/s?wd=地址 |

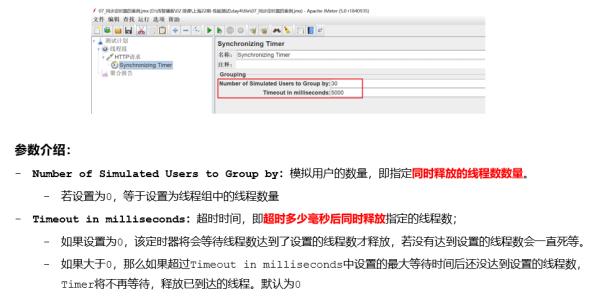


 注意：结束循环字段可以设置为0，自动读取到最后一个

**定时器：**

**同步定时器：**





|  |
| --- |
| 案例：1、模拟100个用户同时访问百度首页，统计各种高并发情况下运行情况 |

使用同步定时器的操作步骤？

1.添加线程组，设置线程数为n

2.添加HTTP请求

3.添加同步定时器

•**设置并发线程数**：同时发送请求的虚拟用户数

•**设置超时时间**：

Ø**建议设置**：不设置的话，若没有达到设置的线程数会一直死等

Ø**不能设置太小**：等待时间后还没达到设置的线程数，会释放已到达的线程

4.添加查看结果树

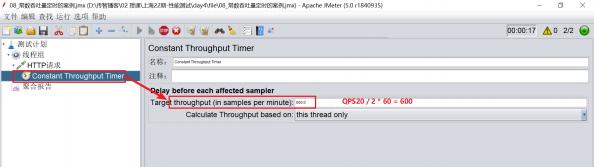
5.添加监听器-聚合报告

**常数吞吐量定时器：**

案例：

（1）一个用户以 20QPS (20 次/s) 的频率访问百度首页，持续一段时间，统计运行情况

（2）2个用户针对 （服务器的QPS要求： 20QPS (20 次/s)） 的频率访问百度首页，持续一段时间，统 计运行情况



**固定定时器：**

案例：

（1） IHRM系统登录错误3次后，锁定1分钟， 1分钟后重新输入正确的用户名密码登录成功

步骤：

 添加线程组

 添加HTTP请求1 -错误1次

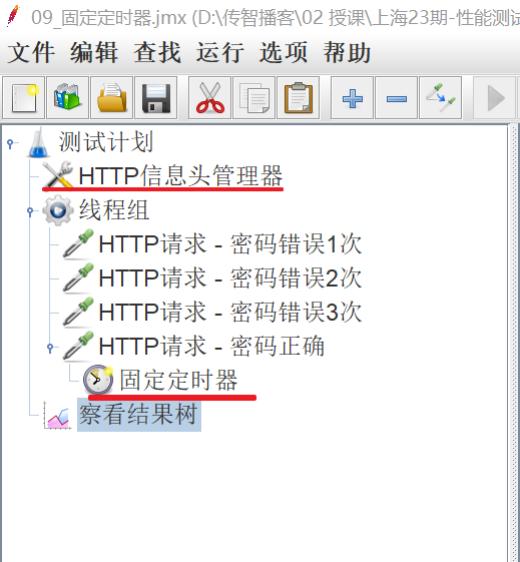
 添加HTTP请求2 -错误2次

 添加HTTP请求3 -错误3次

 添加HTTP请求4 -正确用户名密码

 在HTTP请求4下，添加固定定时器

 添加查看结果树



注意：

 固定定时器，必须添加在需要等待的HTTP请求的子节点下

 在HTTP信息头管理器中，修改HTTP请求的头域

