

性能测试第三天课堂笔记

昨日回顾：

(1) JMeter基本使用

元件：配置元件、前置处理器、定时器、取样器、后置处理器、断言、监听器

作用域：

- 取样器：是核心，本身无作用域
- 逻辑控制器：只对子节点起作用
- 其他元件：
 - 父节点是取样器，只对父节点起作用
 - 父节点不是取样器，对父节点的所有后代节点（子节点、子节点的子节点等）都起作用

执行顺序：

- 同一个作用域的不同元件：配置元件、前置处理器、定时器、取样器、后置处理器、断言、监听器
- 同一个作用域的相同元件：从上到下的顺序

(2) 核心元件

- 线程组：
 - 分类：普通线程组、setup线程组、teardown线程组
 - 属性：线程数、ramp-up时间、循环次数、持续运行时间、延迟启动时间
- HTTP请求
 - GET请求（在URL的路径中）
 - URL：协议、域名、端口、路径、请求方法GET、编码格式
 - 参数：
 - 在路径中添加参数。？参数名 = 参数值&参数名 = 参数值
 - 在参数列表中添加。格式：参数名，参数值
 - POST请求（在消息体数据中）
 - URL：协议、域名、端口、路径、请求方法POST、编码格式
 - 参数：
 - 在参数列表中添加（只能添加form格式）
 - 在消息体数据中添加（任意格式）
- 查看结果树
 - 请求：请求行（URL）、请求体（参数）
 - 响应：状态码、响应体

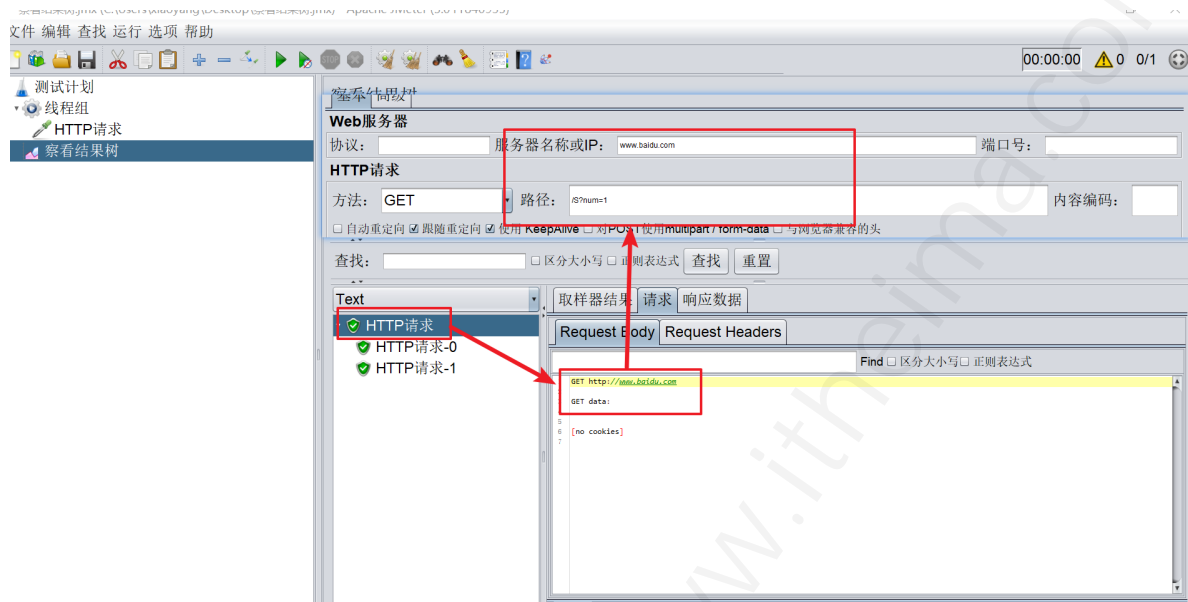
(3) JMeter参数化

作用：使用不同的测试数据，调用相同的测试方法

- 用户定义的变量：定义全局变量
- 用户参数：不同的用户使用不同的测试数据
- CSV数据文件设置：不同的用户，或者同一个用户的多次循环，都可以获取不同的数据
- 函数：不同的用户，或者同一个用户的多次循环，都可以获取不同的数据，不需要提前准备数据。

问题描述:

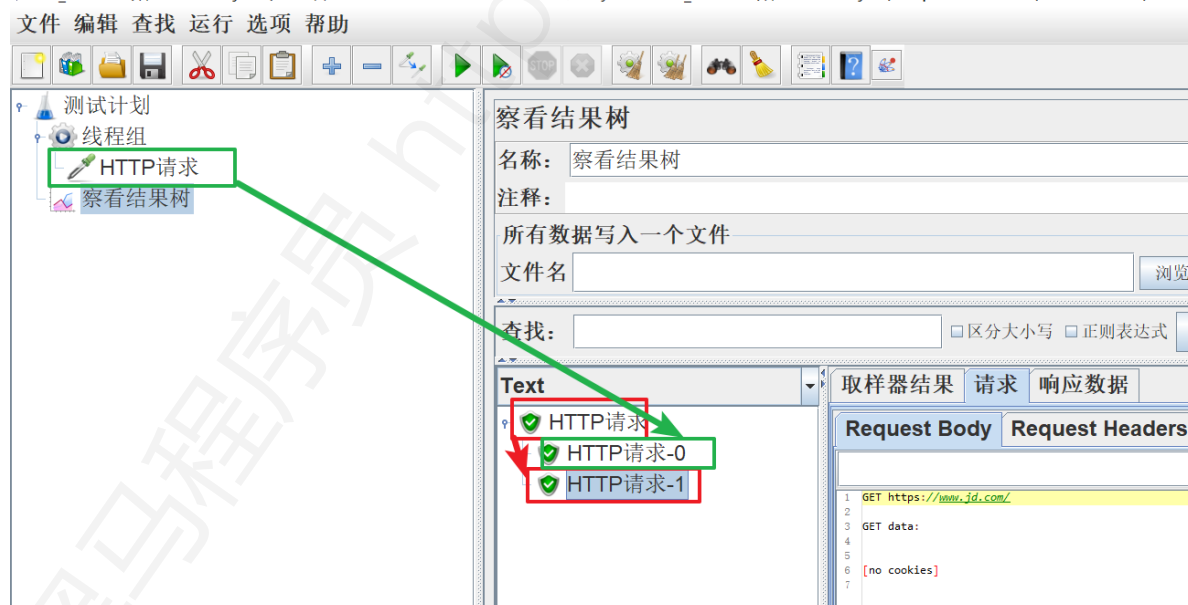
当发送www.jd.com的http请求时，查看结果树看到的发送消息和HTTP取样器中配置的不完全一样？



原因分析:

- 查看结果数中最外层HTTP请求的 请求信息和响应信息，应该与子节点中最后一个HTTP请求的请求消息和响应消息一致；
- 配置的HTTP请求，应该与子节点中第一个HTTP请求的请求信息一致

01_查看结果数的错误案例.jmx (D:\传智播客\02 授课\上海23期-性能测试\day3\files\01_查看结果数的错误案例.jmx) - Apache JMeter (5.0 r1840935)



- JMeter可以自动判断/处理响应状态码

昨日作业:

某支付系统，需要用1000个不同的用户登录，并使用添加不同的测试金额数据访问支付接口？

更多资源，百度“黑马软件测试学习路线图” bbs.itheima.com/thread-405757-1-1.html

答案:

- 1、准备CSV数据文件 —— 1000个登录的用户名密码
- 2、添加线程组，设置线程数为1000
- 3、添加**CSV数据文件设置**
- 4、添加HTTP请求 - 登录，使用读取的用户名密码
- 5、添加HTTP请求 - 支付 —— 使用**counter函数**添加不同的测试金额
- 6、查看结果树

今日重点:

- 掌握Jmeter断言使用
- 掌握Jmeter关联使用
- 掌握Jmeter自动录制脚本

性能测试工具

Jmeter断言:

断言：让程序自动判断预期结果和实际结果是否一致。

提示:

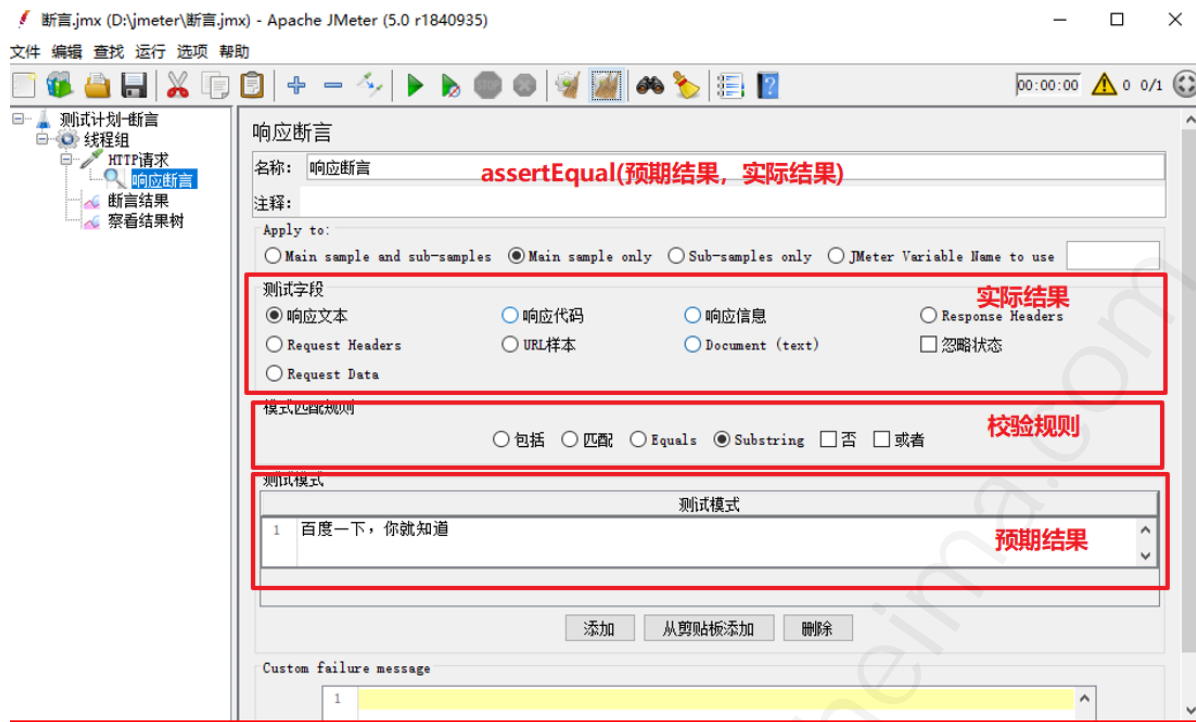
- JMeter在请求的返回层面有个自动判断机制（响应状态码）
- 但是请求成功了，并不代表结果一定正确，因此需要检测机制提高测试准确性。

JMeter中常用断言:

- 响应断言
- JSON断言
- 持续时间断言 (Duration Assertion)

响应断言:

参数:



参数配置详细介绍:

测试字段: 要检查的项 (实际结果)

- 响应文本: 来自服务器的响应文本, 即主体
- 响应代码: 响应的状态码, 例如: 200
- 响应信息: 响应的信息, 例如: OK
- Response Headers: 响应头部
- Request Headers: 请求头部
- Request Data: 请求数据
- URL样本: 请求URL
- Document (text): 响应的整个文档
- 忽略状态: 忽略返回的响应状态码

模式匹配规则: 比较方式

- 包括: 文本包含指定的正则表达式
- 匹配: 整个文本匹配指定的正则表达式
- Equals: 整个返回结果的文本等于指定的字符串 (区分大小写)
- Substring: 返回结果的文本包含指定字符串 (区分大小写)
- 否: 取反
- 或者: 如果存在多个测试模式, **勾选代表逻辑或** (只要有一个模式匹配, 则断言就是OK), **不勾选代表逻辑与** (所有都必须匹配, 断言才是OK)
- **注意: Equals和Substring模式是普通字符串, 而包括和匹配模式是正则表达式**

测试模式: 预期结果

- 即填写你指定的结果 (可填写多个), 按钮【添加】、【删除】是进行指定内容的管理

案例1: 检查百度首页的接口响应中包含“百度一下, 你就知道”

1. 添加线程组

2. 添加HTTP请求

3. 添加响应断言

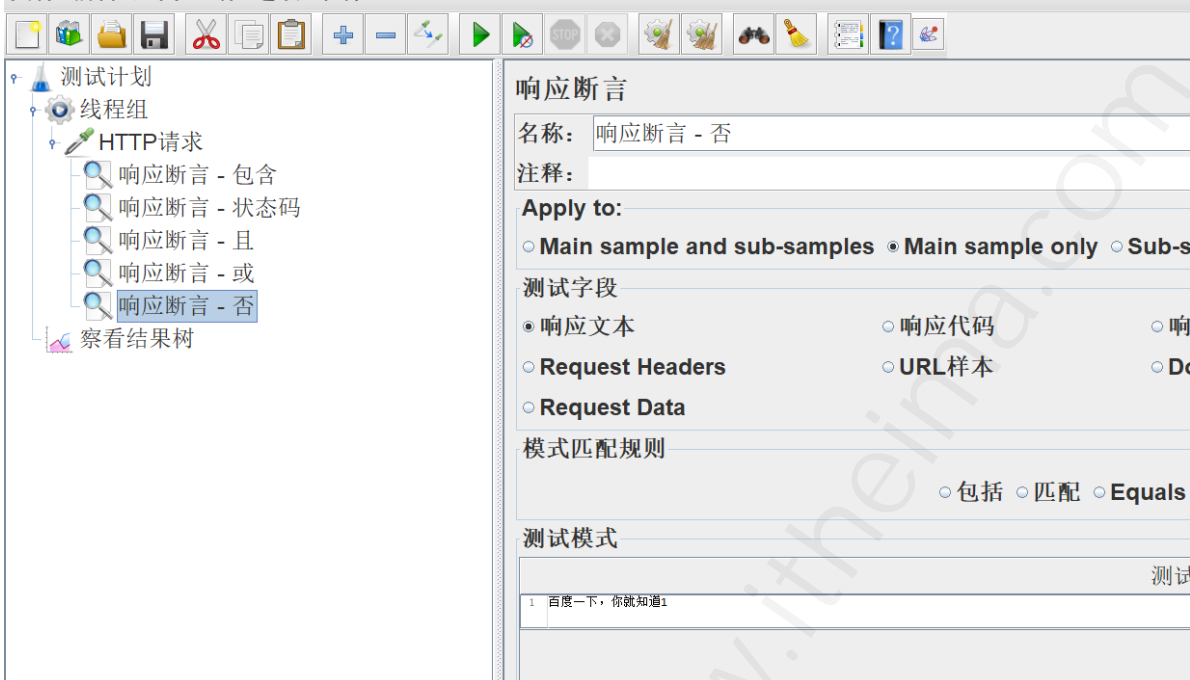
- 测试字段: 要检查的项 (实际结果)
- 模式匹配规则: 比较方式
- 测试模式: 预期结果

4. 添加查看结果树

案例2：响应断言中，有多个待校验的测试数据

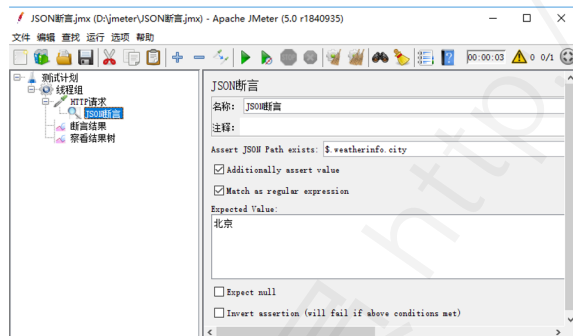
02_响应断言.jmx (D:\传智播客\02 授课\上海23期-性能测试\day3\files\02_响应断言.jmx) - Apache JMeter (5.0 r1840935)

文件 编辑 查找 运行 选项 帮助



JSON断言：

参数：



参数介绍：

- **Assert JSON Path exists**: 用于断言的JSON元素的路径 (**实际结果**)
- **Additionally assert value**: 如果您想要用某个值生成断言，请选择复选框
- **Match as regular expression**: 使用正则表达式断言
- **Expected Value**: 期望值 (**期望结果**)
- **Expect null**: 如果希望为空，请选择复选框
- **Invert assertion (will fail if above conditions met)**: 反转断言 (如果满足以上条件则失败)

案例：

请求: <http://www.weather.com.cn/data/sk/101010100.html>

检查: 让程序检查响应的JSON数据中，city对应的内容是否为“北京”

(1) 什么时候可以使用JSON断言?

- 对HTTP请求的响应结果为JSON格式时, 可以使用JSON断言

(2) 使用“JSON断言”的操作步骤?

1. 添加线程组

2. 添加HTTP请求

3. 添加JSON断言

- 填写Assert JSON Path exists (**实际结果-json路径**)
- 勾选Additionally assert value
- 填写Expected Value (**期望结果**)

4. 添加查看结果树

断言持续时间:

作用: 检查HTTP请求的响应时间是否超出要求范围

位置: 测试计划 --> 线程组--> HTTP请求 --> (右键添加) 断言 --> 断言持续时间



参数介绍:

- 持续时间 (毫秒): HTTP请求允许的**最大响应时间** (**单位: 毫秒**)。 超过则认为失败

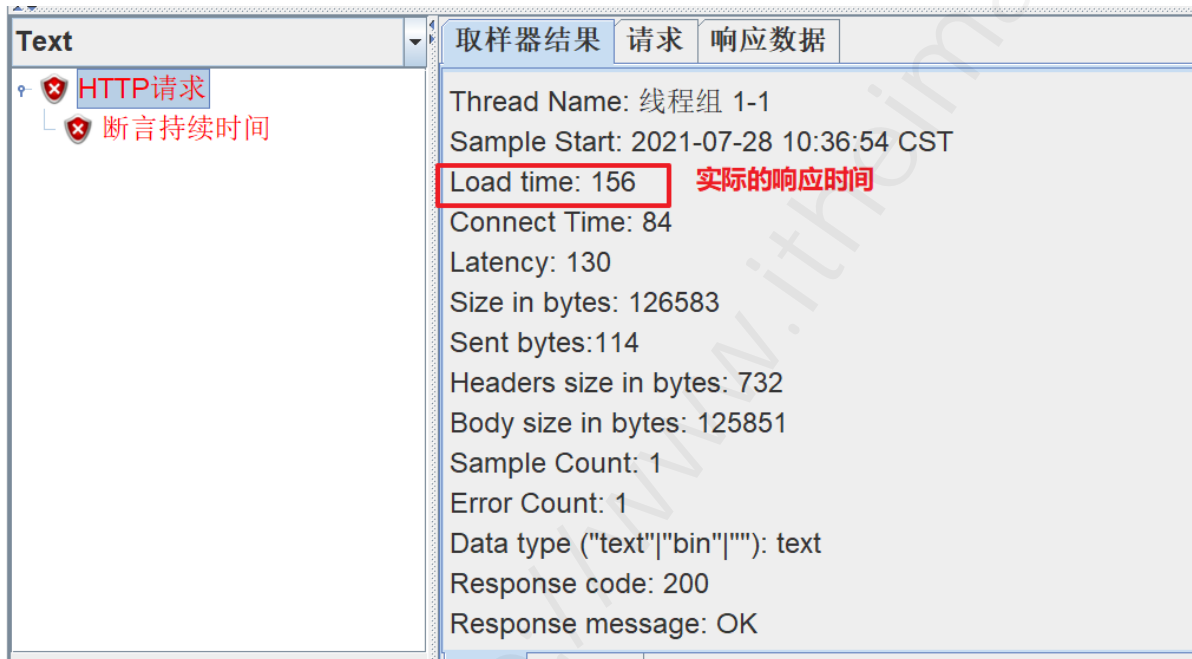
案例:

请求: <https://www.jd.com>

检查: 让程序检查响应时间是否大于100毫秒

(2) 使用“断言持续时间”的操作步骤？

1. 添加线程组
2. 添加HTTP请求
3. 添加断言持续时间
 - 填写持续时间（允许的最大响应时间，单位：ms）
4. 添加查看结果树



Jmeter关联：

关联：当请求之间有**依赖关系**，比如一个请求的入参是另一个请求返回的数据，这时候就需要用到关联处理。

JMeter中常用的关联方法：

- 正则表达式提取器
- XPath提取器
- JSON提取器

正则表达式介绍：

正则表达式：就是一个公式，或者说一套规则，使用这套规则可以从任意字符串中提取出想要的数据内容

公式格式：左边界(匹配符号)右边界：可以提取出想要获取的数据内容

.：是通配符，可以代表任意字符（除换行回车）

*：代表前面的字符出现0次或者多次

. *匹配规则：找到左边界值后，往右查找有边界，找到最后面的右边界，中间的所有数据都被记录下来

?：代表非贪婪匹配，找到左边界后，往右查找匹配右边界，只要有匹配的右边界就停止继续查找；再次查找左边界和右边界

公式格式：左边界(. *?)右边界

<title>百度一下，你就知道</title><title>百度一下，你就知道</title>

<title>(. *?)</title>

案例二：

原始文本：

021-1234-1234

022-1234-1235

023-1234-1236

024-1234-1237

025-1234-1238

026-1234-1239

027-1234-1230

要求：匹配出 城市号、地区号、个人号码三组

正则表达式匹配：

(. *?)-(.*?)-(.*?)\n

结论：

- 通过一个正则表达式可以提取出多组数据，每组数据设置对应的左边界和右边界即可
- 每一组数据都可以有一个或者多个值

正则表达式提取器：

(1) 什么时候可以使用正则表达式提取器?

- 任意格式的响应数据, 都可以使用正则表达式提取器进行提取

(2) 使用“正则表达式提取器”的操作步骤?

1. 添加线程组

2. 添加HTTP请求-传智播客

3. 添加正则表达式提取器

- 引用名称: 存放提取出的值的**参数名称**, 如填写title
- 正则表达式: **左边界(.*)右边界**
- 模板: **用\$\$引用起来**, 表示解析出第几个()的值
- 匹配数字: **1表示第1个值, -1表示所有取值**

4. 添加HTTP请求-百度

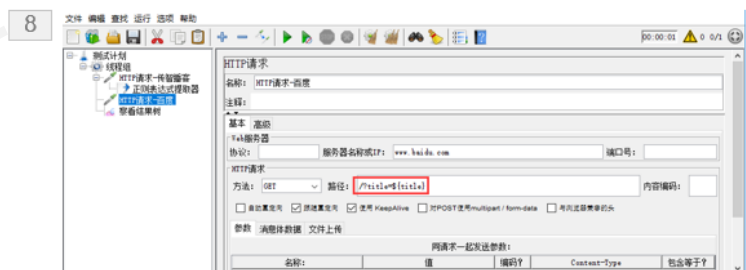
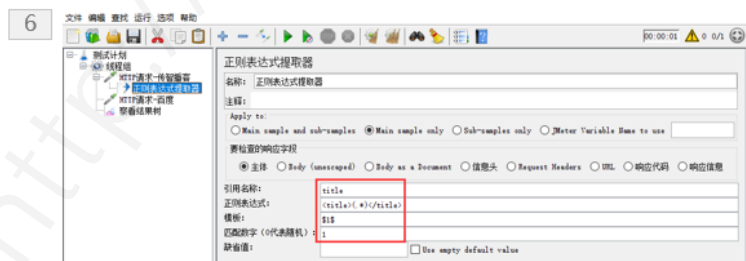
- 引用正则表达式中的引用名称。如: 用\${title}引用它

5. 添加查看结果树

案例1: 获取传智播客首页的title, 并作为参数传递

操作步骤:

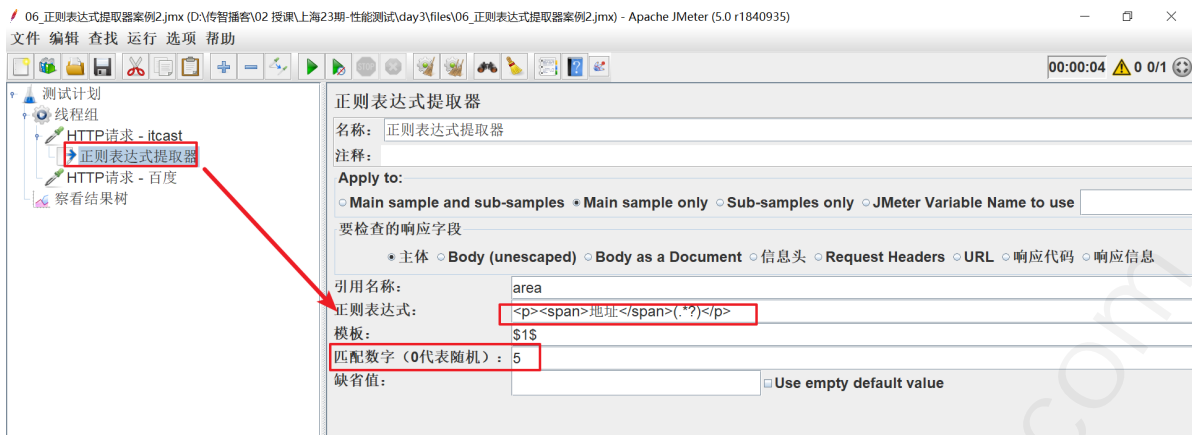
- 添加线程组
- 添加HTTP请求-传智播客
- 添加正则表达式提取器
- 添加HTTP请求-百度
- 添加查看结果树



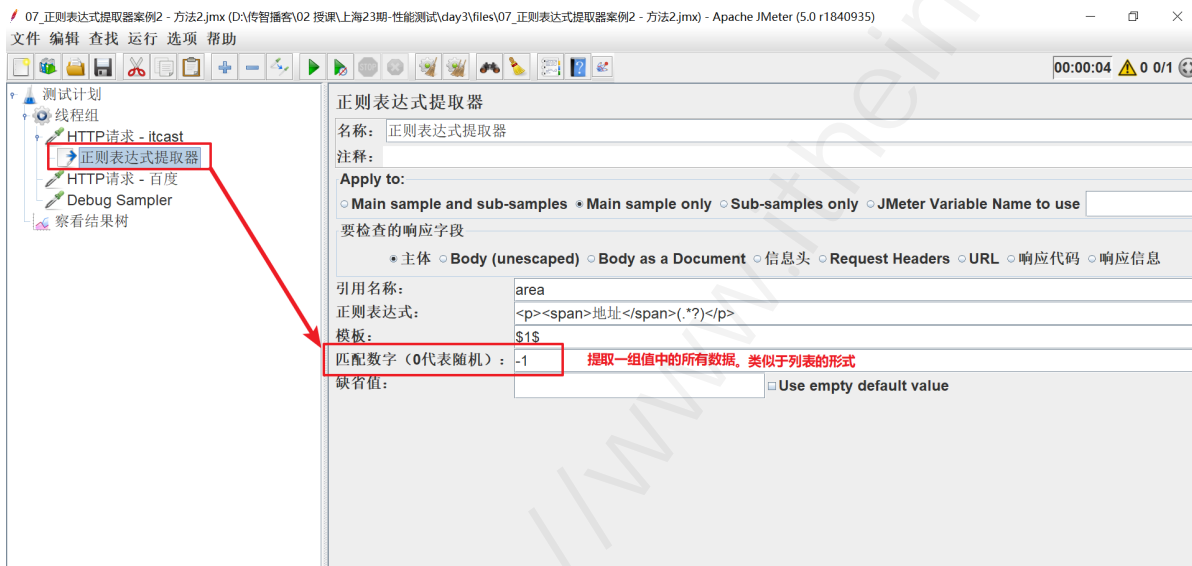
案例2: 获取传智播客首页的地址, 把第5个校区地址作为参数传递 (span为地址的校区)

响应中的地址格式显示: <p>地址上海市浦东新区航头镇航都路18号万香创新港</p>

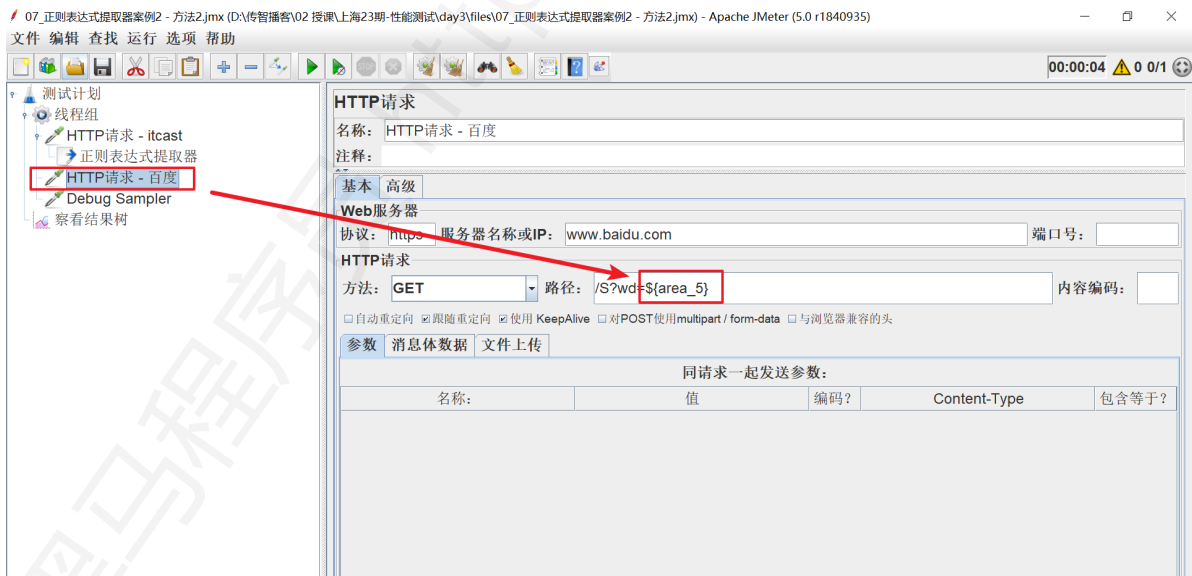
方法1:



方法2:



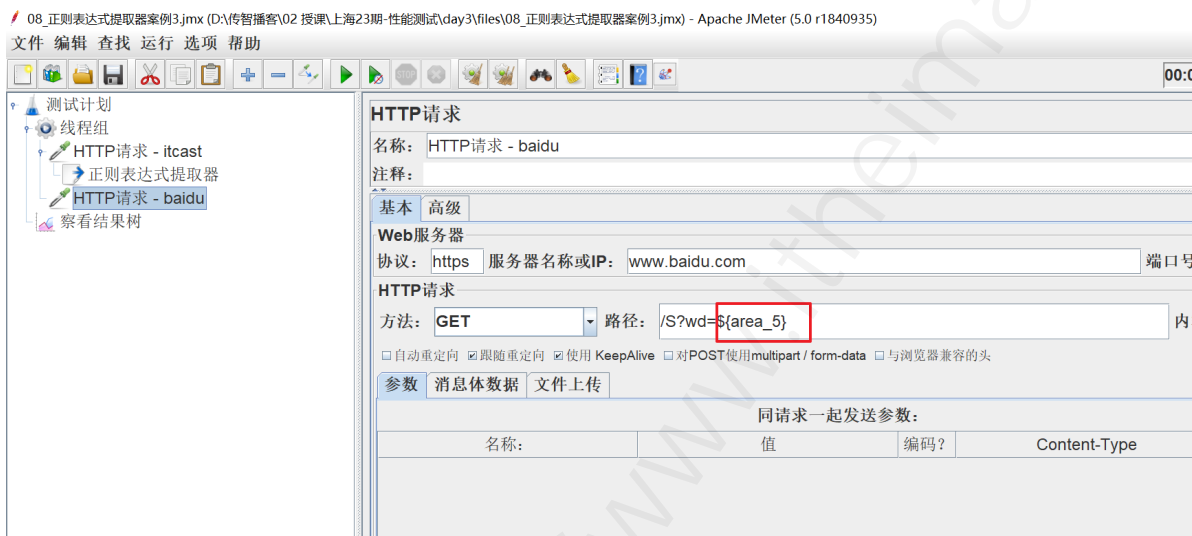
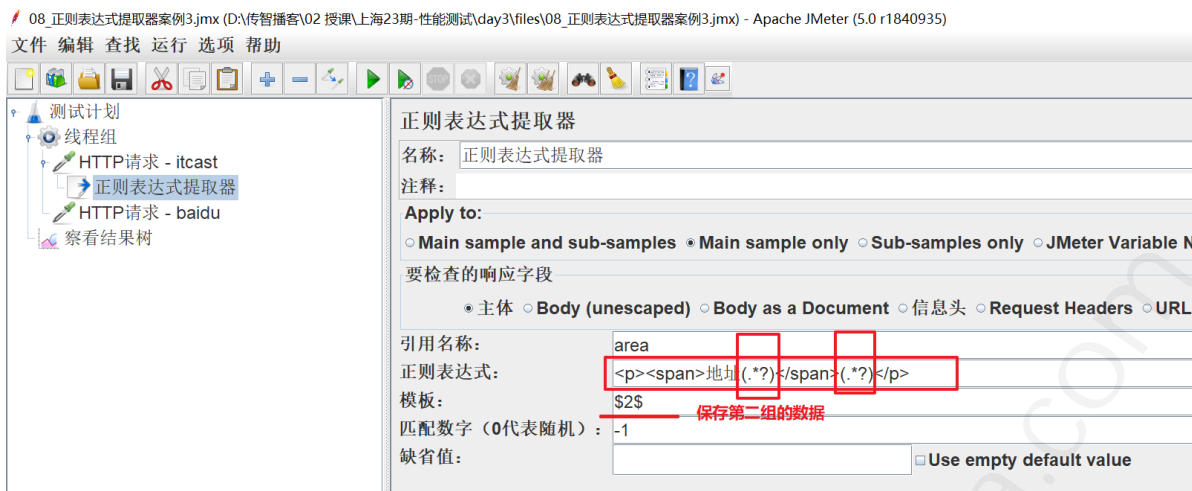
- 引用：以列表索引的方式来引用。格式：\${变量名_索引}，索引从1开始



案例3：获取传智播客首页的地址，把第5个校区地址作为参数传递（span为地址/后面带一、二、三）

地址格式1: <p>地址一昌平区建材城西路金燕龙办公楼一层</p>

地址格式2: <p>地址上海市浦东新区航头镇航都路18号万香创新港</p>



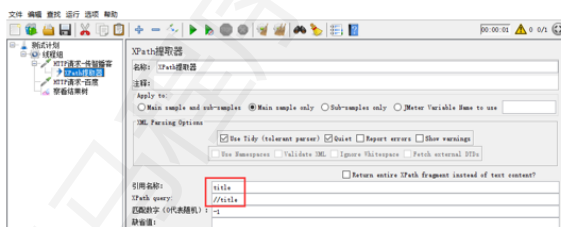
xpath提取器:

案例:

//a[@id='kw']: 在HTML页面中, 找出a标签 (有一个属性为id, 且id的值为kw)
//a : 找出所有的a标签

作用: 针对HTML格式的响应结果数据进行提取

位置: 添加方式: 测试计划 --> 线程组 --> HTTP请求 --> (右键添加) 后置处理器 --> XPath提取器



参数介绍:

- Use Tidy (tolerant parser):**
 - 当需要处理的页面是HTML格式时, 必须选中该选项
 - 当需要处理的页面是XML或XHTML格式时, 取消选中该选项。
- 引用名称:** 存放提取出的值的参数名称
- XPath Query:** 用于提取值的XPath表达式
- 匹配数字:** 如果XPath路径查询出许多结果, 则可以选择提取哪个。
0: 表示随机, -1: 表示提取所有结果, 1表示第一个值
- 缺省值:** 参数的默认值

案例: 获取传智播客首页的title, 并作为参数传递

1. 添加线程组
2. 添加HTTP请求-传智播客
3. 添加XPath提取器

- **勾选Use Tidy (tolerant parser)**
- **引用名称:** 存放提取出的值的**参数名称**。如: 填写title
- **XPath Query:** 用于提取值的**XPath表达式**
- **匹配数字:** 0: 表示随机, **-1: 表示提取所有结果, 1表示第一个值**

4. 添加HTTP请求-百度

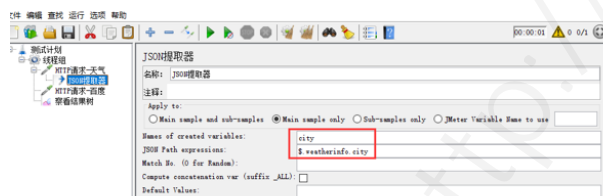
- 引用正则表达式中的引用名称。如: 用**`${title}`**引用它

5. 添加查看结果树

json提取器:

作用: 针对JSON格式的响应结果数据进行提取

位置: 添加方式: 测试计划 --> 线程组 --> HTTP请求 --> (右键添加) 后置处理器 --> JSON提取器



参数介绍:

- **Names of created variables:** 存放提取出的值的参数名称
- **JSON Path Expressions:** JSON路径表达式
- **Match No:** 如果JSON路径匹配出许多结果, 则可以选择提取哪个。
0: 表示随机, -1: 表示提取所有结果, 1表示第一个值
- **Default Values:** 参数的默认值

案例: 提取天气接口中的城市信息, 作为参数在访问百度首页时传递

1. 添加线程组
2. 添加HTTP请求-天气
3. 添加JSON提取器

- **Names of created variables:** 存放提取出的值的**参数名称**。如: cit
- **JSON Path Expressions:** 用于提取值的**JSON路径表达式**
- **Match No:** 0: 表示随机, **-1: 表示提取所有结果, 1表示第一个值**

4. 添加HTTP请求-百度

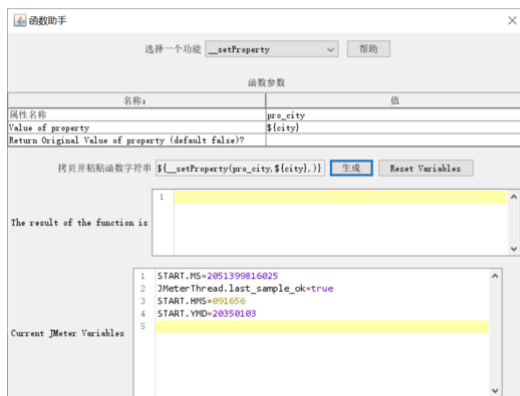
- 引用正则表达式中的引用名称。如: 用**`${city}`**引用它

5. 添加查看结果树

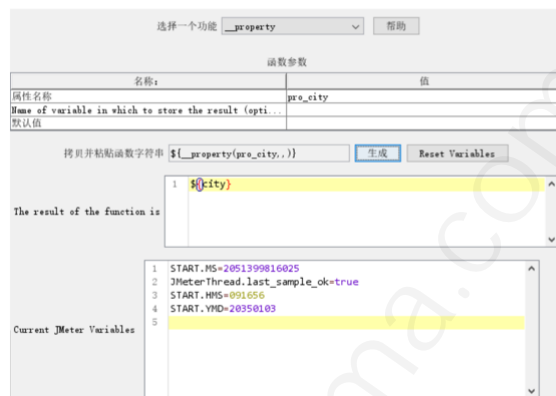
JMeter属性:

JMeter属性的配置函数:

__setProperty函数: 将值保存成JMeter属性

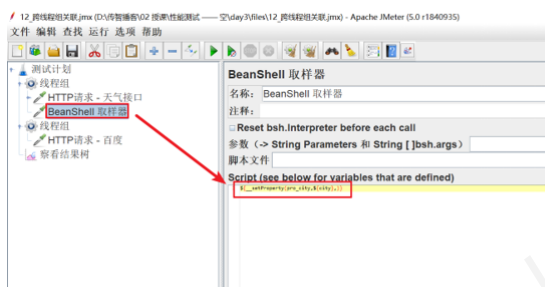


__property函数: 在其他线程组中使用property函数读取属性

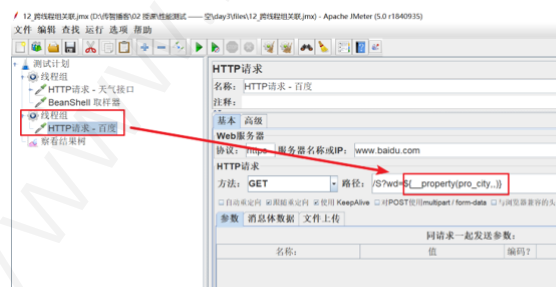


JMeter属性的函数执行:

__setProperty函数执行 (保存JMeter属性):
— 需要通过BeanShell取样器来执行



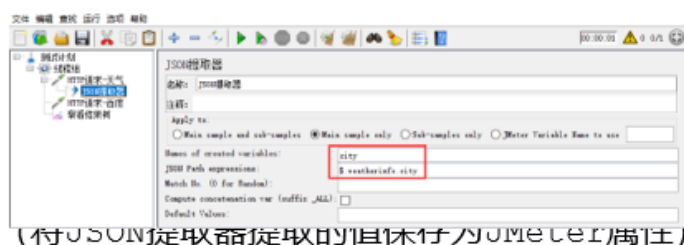
__property函数 (读取属性):
— 在其他线程组中使用property函数



案例:

- 1、线程组1: 请求获取天气的接口, <http://www.weather.com.cn/data/sk/101010100.html>, 获取返回结果中的城市名称“北京”**
- 2、线程组2: 请求<https://www.baidu.com/s?wd=北京>, 把获取到的城市名称作为请求参数**

1. 添加线程组1
2. 添加HTTP请求-天气
3. 添加JSON提取器
4. 添加BeanShell取样器



5. 添加线程组2
6. 添加HTTP请求-百度 (读取JMeter属性)
 - 读取JMeter属性: `__property('pro_city',)`
7. 添加查看结果树

自动录制脚本

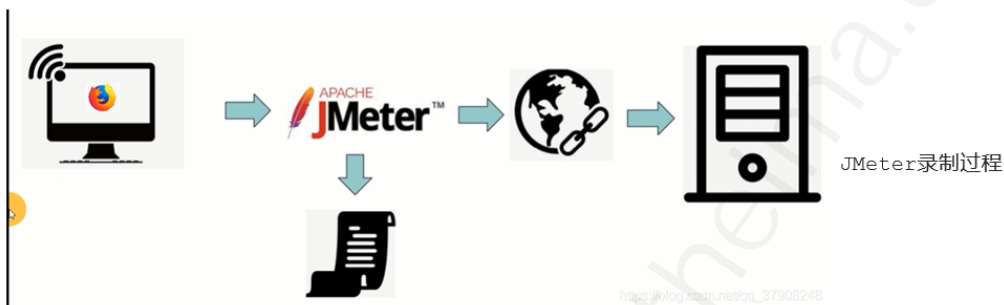
原理：

JMeter录制脚本：在**没有接口文档的旧项目**当中，快速录制web页面产生的http接口请求，**帮助编写接口测试脚本**。

JMeter录制脚本原理：



录制时，JMeter作为**代理服务器**来拦截和转发请求与响应数据

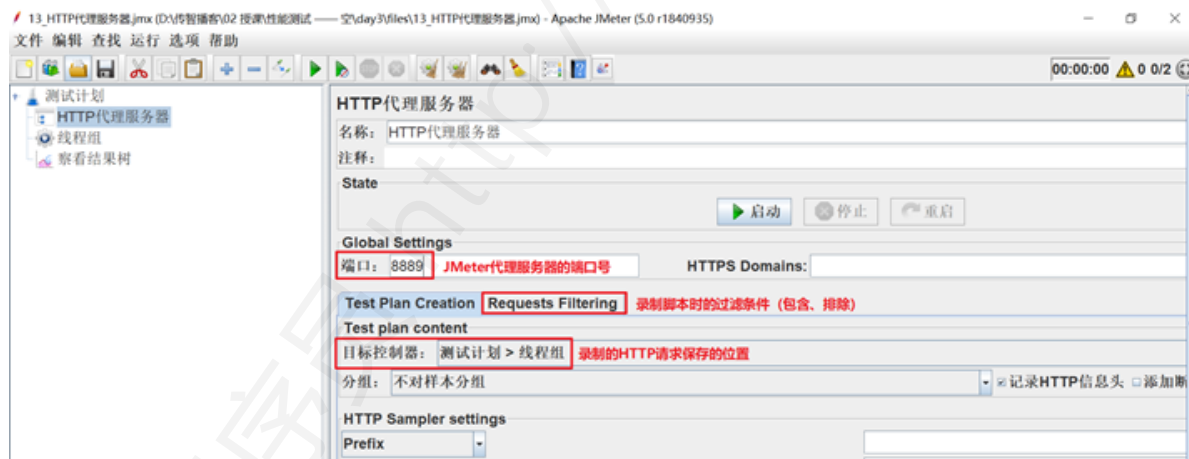


jmeter脚本录制：

1.添加HTTP代理服务器，并进行配置

•加HTTP代理服务器：测试计划(右键)->非测试元件->HTTP代理服务器

•配置代理服务器的参数



2. 开启windows操作系统的浏览器代理

手动设置代理

将代理服务器用于以太网或 Wi-Fi 连接。这些设置不适用于 VPN 连接。

使用代理服务器

☒ 开 **1、打开开关**

地址: 127.0.0.1 端口: 8889 **2、jmeter代理服务器的IP+port**

请勿对以下条目开头的地址使用代理服务器。若有多个条目，请使用英文分号 (,) 来分隔。

<-loopback> **3、可选，当服务器在本机时，建议配置<-loopback>**

☐ 请勿将代理服务器用于本地(Intranet)地址 **4、抓取本机网络报文时不能勾选**

保存 **5、保存**

3. 启动代理服务器，开始录制

HTTP代理服务器

名称: HTTP代理服务器

注释:

State

Global Settings

端口: 8889 HTTPS Domains:

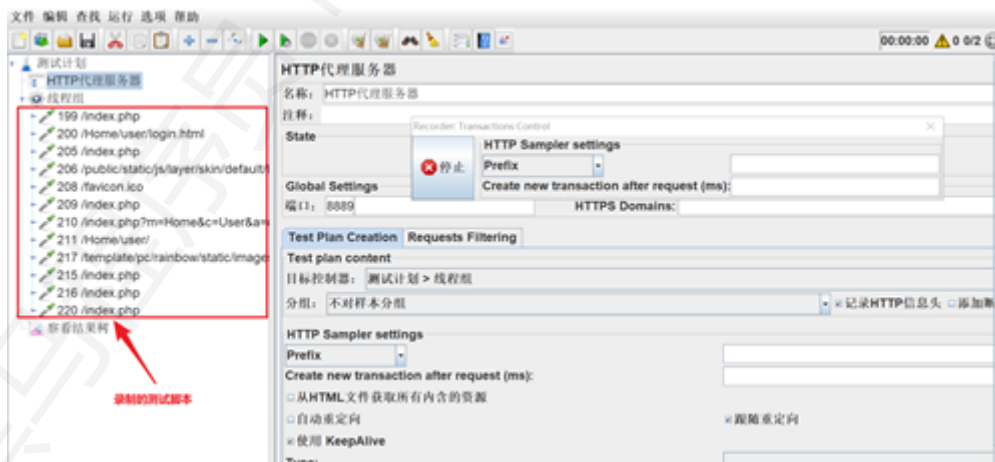
Recorder: Transactions Control

HTTP Sampler settings

Prefix

Create new transaction after request (ms):

4. 在浏览器页面中进行操作，成功后，就能在JMeter当中看到抓取到的接口请求了。



练习:

设置过滤规则，录制tpshop的登录脚本，运行脚本观察测试结果

当使用代理的过程中，发现抓不到包，几个可能的情况：

- 过滤规则设置有问题
- 重启Jmeter代理服务器或者重启Jmeter
- 换浏览器来使用（Chrome、IE）
- 检查PC机中的代理设置是否处于可用状态
- 拔掉网线，抓包