###### Jmeter接口测试（上）

1、Jmeter工作目录

bin：放置各项配置文件（如日志设置、JVM设置）、启动文件、启动jar包、示例脚本。

docs：放置jmeter API的离线帮助文档。

extras：jmeter辅助功能，提供与Ant、Jenkins集成的可能性，

用来构建性能测试自动化框架。

lib：jmeter组件已jar包的形式放置在lib/ext目录下，如果要扩展jmeter组件，

jar包就放在此目录下，jmeter启动时会自动加载此目录下的jar包。

printable\_docs：放置jmeter的离线帮助文件，可用来学习jmeter。

2、JMeter主要组件

1. 测试计划(Test Plan)是使用 JMeter 进行测试的起点，是其它 JMeter 测试元件的容器。
2. 线程组(Thread Group)代表一定数量的并发用户，它可以用来模拟并发用户发送请求。
3. 取样器(sampler)

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。

1. 监听器(Listener)负责收集测试结果，同时也被告知了结果显示的方式。
2. 逻辑控制器(Logic Controller)

自定义jmeter发送请求的行为逻辑，他与Sampler结合使用可以模拟复杂的请求序列。

1. 断言(Assertions)

可以用来判断请求响应的结果是否如用户所期望的。它可以用来隔离问题域，

即在确保功能正确的前提下执行压力测试。这个限制对于有效的测试是非常有用的。

1. 配置元件(Config Element)

维护Sampler需要的配置信息，并根据实际的需要会修改请求的内容。

1. 前置处理器(Pre Processors)和后置处理器(Post Processors)

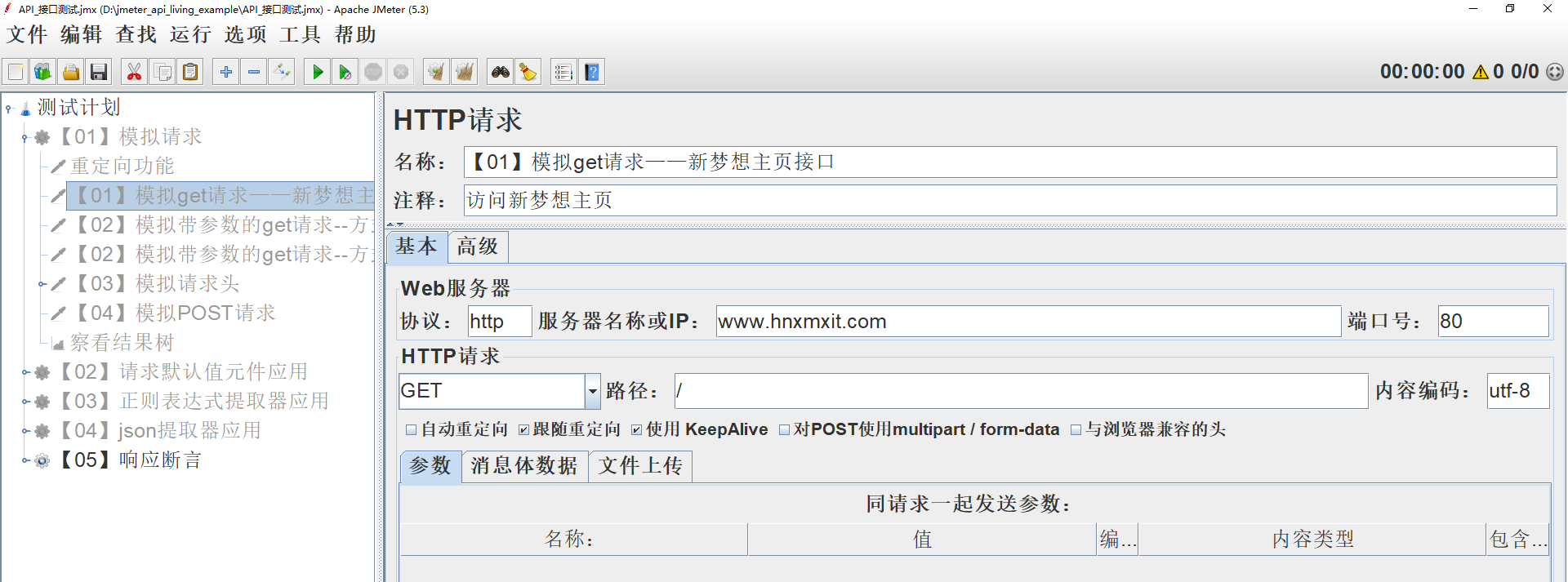
负责在生成请求之前和之后完成工作。

前置处理器常常用来修改请求的设置，后置处理器则常常用来处理响应的数据。

1. 定时器(Timer)负责定义请求之间的延迟间隔。

3、Jmeter取样器HTTP请求配置说明

取样器是用来模拟用户操作，向服务器发送请求以及接收服务器的响应数据的一类元件，其中HTTP请求取样器是用来模拟常用的http请求的。



基本界面介绍：

1. 名称：本属性用于标识一个取样器，建议使用一个有意义的名称。
2. 注释：对于测试没有任何作用，仅用户记录用户可读的注释信息。
3. 协议：向目标服务器发送HTTP请求时的协议，可以是http或者是https或者File ，

默认值为http 。

1. 服务器名称或IP ：HTTP请求发送的目标服务器名称或IP地址

注意：此处不可带http:/或者https:/

1. 端口号：目标服务器的端口号，默认值为80 。
2. 方法：发送HTTP请求的方法，可用方法包括GET、POST等。
3. 路径：目标URL路径（不包括服务器地址和端口）-也可以直接写完整路径。
4. 内容编码 ：内容的编码方式，默认值为iso8859

一般设置为utf-8，防止get参数或者post参数有中文会乱码情况。

1. 自动重定向：如果选中该选项，当发送HTTP请求后得到的响应是302/301时，

JMeter 会自动重定向到新的页面，但是Jmeter是不记录重定向的过程内容。

1. 跟随重定向：Http Request取样器的默认选项，

当响应code是3xx时，自动跳转到目标地址。

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，

如有多个跳转则多个请求都会被记录下来，此为默认选项。

自动重定向和跟随重定向的区别只在于是否记录多个跳转的请求上

11) 使用 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。

12) 对POST使用multipart/from-data ：

当发送HTTP POST请求时，使用Use multipart/from-data方法

或者application/x-www-form-urlencoded方法发送，默认不选中。

1. 与浏览器兼容的头：

当勾选multipart/form-data时，勾选此项会截掉http请求头中的Content-Type

和Content-Transfer-Encoding，而只发送Content-Disposition部分；

1. 参数 ：

可以将URL中所有参数设置在本表中，表中的每一行是一个参数值对。内容编码选项最好勾选，防止参数值内含有ASCII Control Chars或者Non-ASCII characters或者其他符号导致乱码问题。

1. 文件上传：

同请求一起发送的文件信息，参数名称就是发送文件对应的参数，

MINE类型为资源媒体类型。

1. 消息体数据：同请求一起发送的数据信息，可以支持多种数据格式，json、xml等
2. 客户端实现：默认为：Httpclient

Httpclient：java工具包实现请求==基于net工具包 二次定制 效率

Java：jdk自带的net工具包实现请求==原生客户端请求实现

1. 超时（毫秒）

连接：连接超时时间，单位为毫秒；   
 响应：响应等待超时时间，单位为毫秒；

1. 从HTML文件获取所有内含的资源：

当该选项被选中时，jmeter在发出HTTP请求并 获得响应的HTML文件内容后，

还对该HTML进行解析 并获取HTML中包含的所有资源（图片、flash等），

默认不选中

1. 并行下载：设置是否使用自设资源池，勾选后可设置大小。
2. 数量：资源池大小，默认为6
3. 源地址：只用于http协议且Implementation为HttpClient4的情况。

此属性用于启用IP欺骗。会重写了这个http请求使用的默认本地IP地址。

用于Jmeter主机具有多个IP地址（即IP别名、网络接口、设备）的情况。

该值可以是主机名、IP地址或网络接口设备，如“ey0”或“l0”或“wlan0”。

1. 代理服务器：

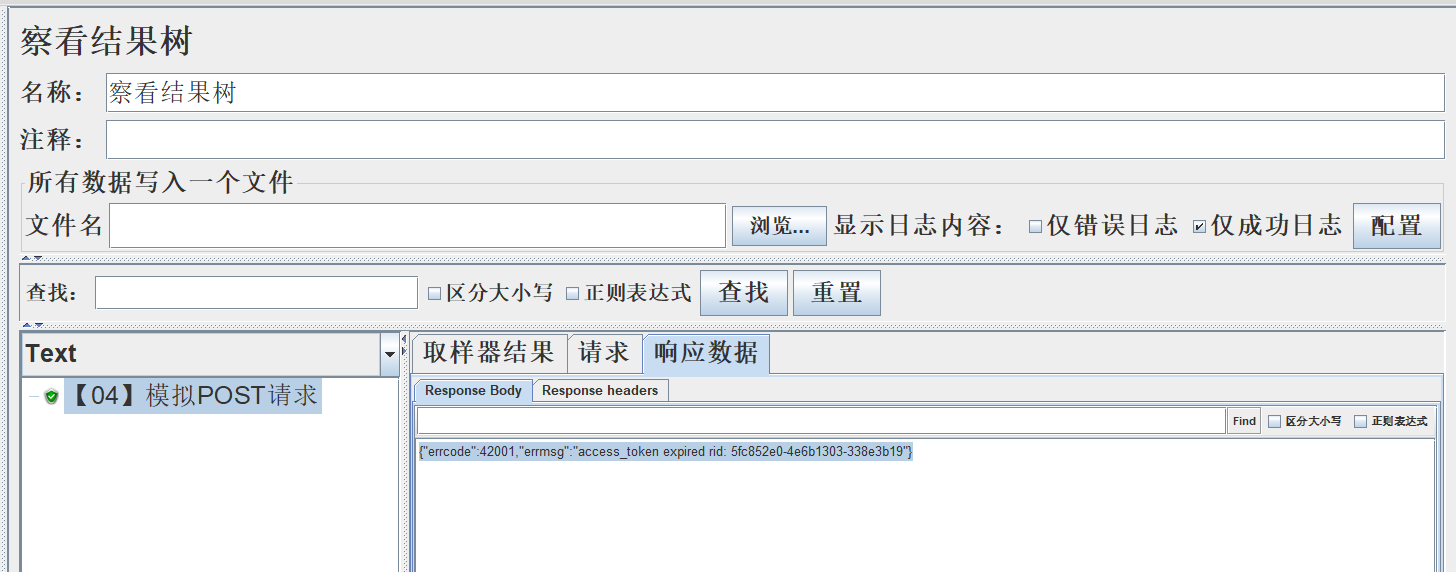
服务器名称或IP：代理服务器的名称或者IP地址；

端口号：该代理的端口号；

用户名：使用该代理的用户名；

密码：用户密码；

1. jmeter察看结果树



1. 文件名：本属性用于标识一个察看结果树元件，建议使用一个有意义的名称。
2. 所有数据写入一个文件：

文件名（\*.jtl）：可以通过浏览选择一个文件，这样jmeter在执行的过程中，

会将所有的信息输出到文件，也支持打开一个结果文件进行浏览。

显示日志内容：

a、仅日志错误：表示只输出报错的日志信息；

b、仅成功日志：表示只输出正常响应的日志信息；

c、在不勾选时，表示输出所有的信息

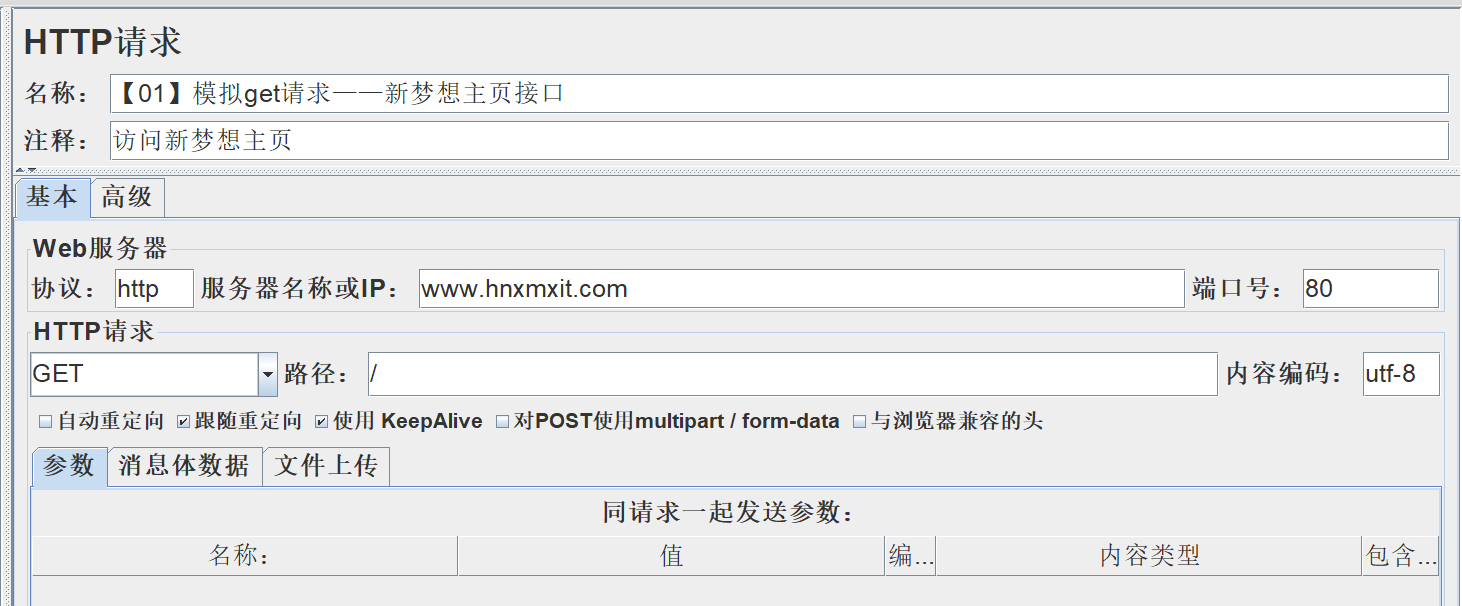
3) 配置：配置需要输出的内容

4) 查找：在输入框中输入想查询的信息，点击Search，可以在请求列表中进行查询，

并在查询出的数据上加上红色的边框。点击重置后，会清除数据上的红色边框。

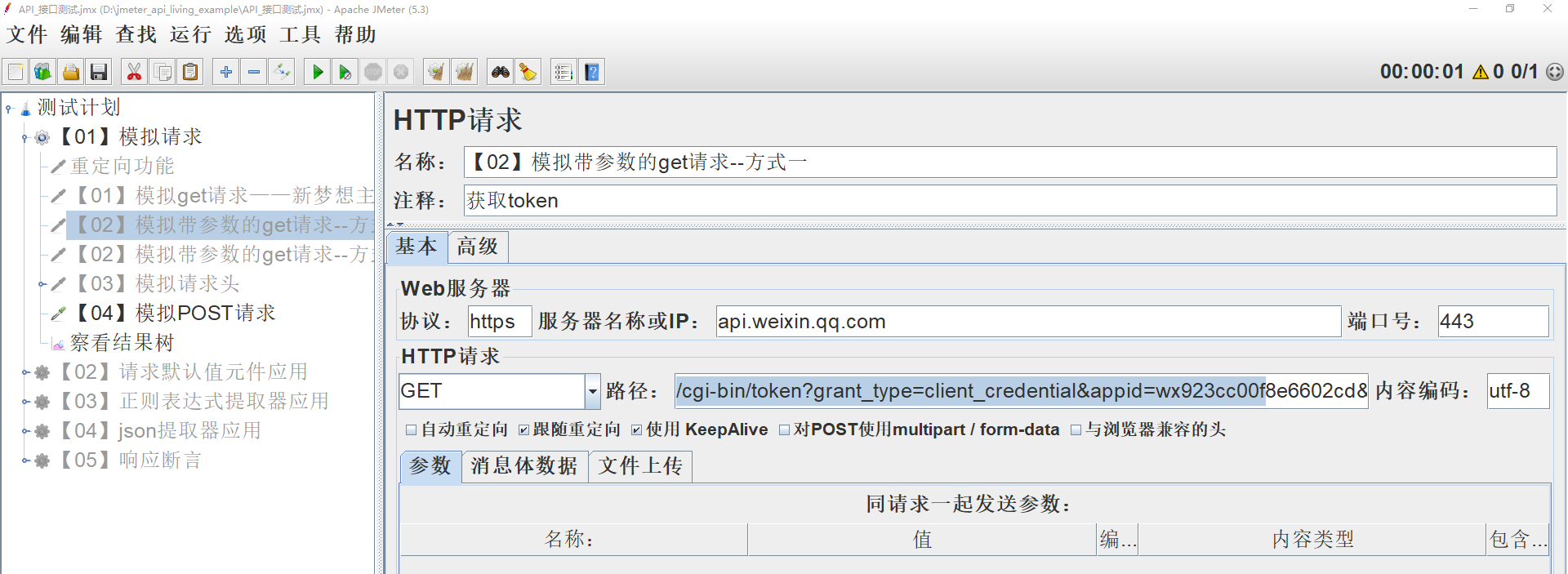
区分大小写和正则表达式，勾选后针对查找生效

5、jmeter模拟get请求

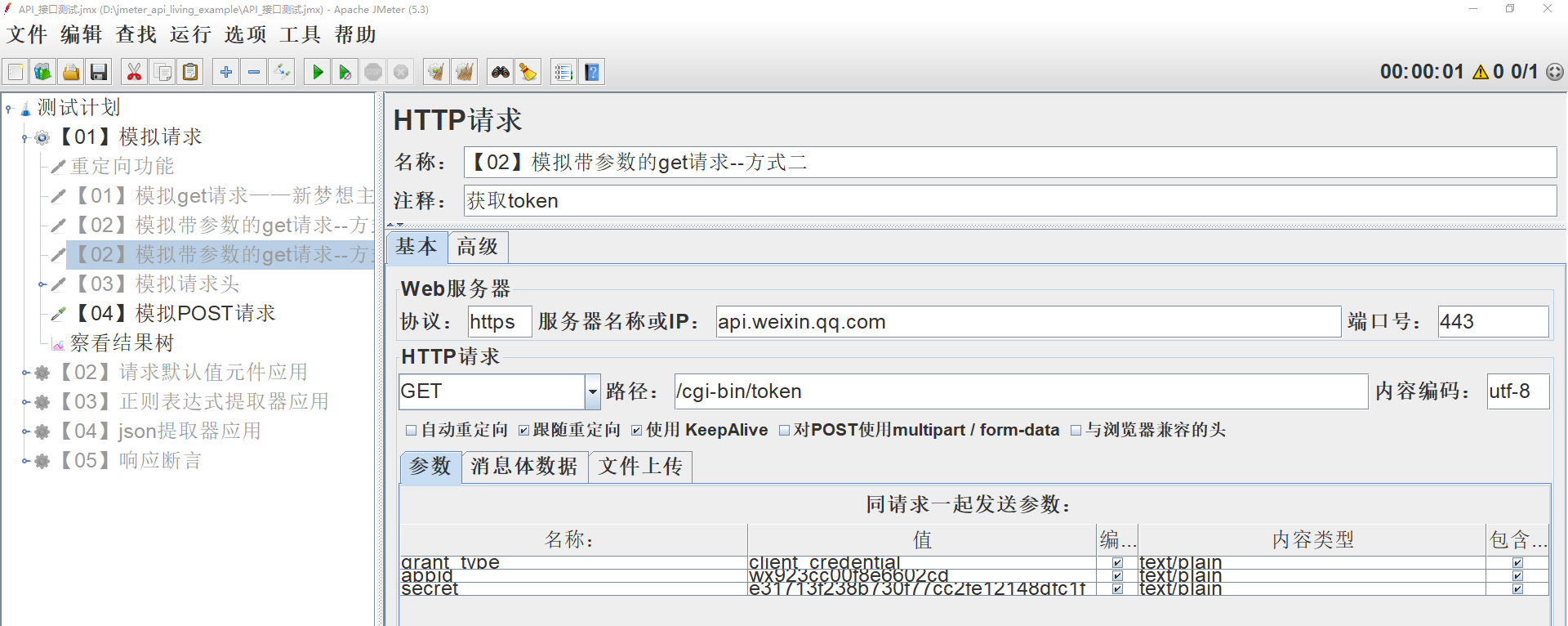


6、jmeter模拟带参数的get请求

方式一：

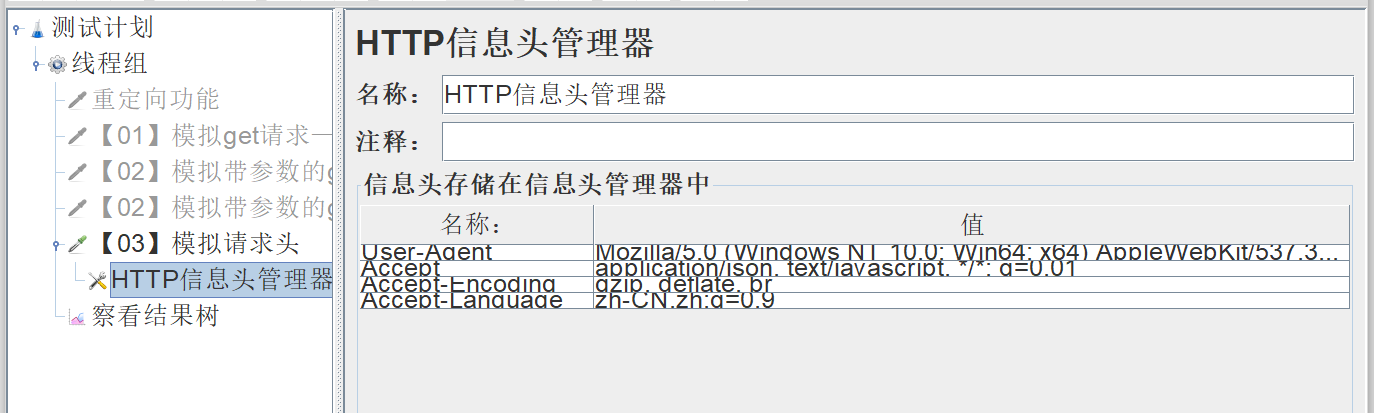


方式二：

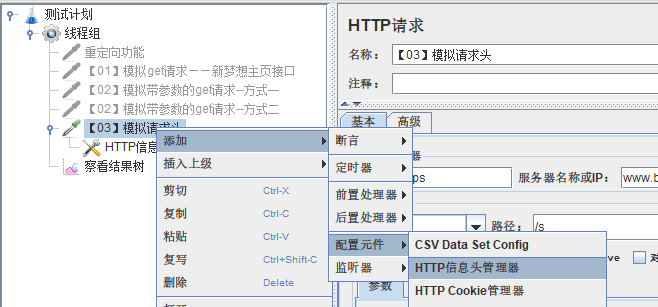


7、Jmeter模拟请求头



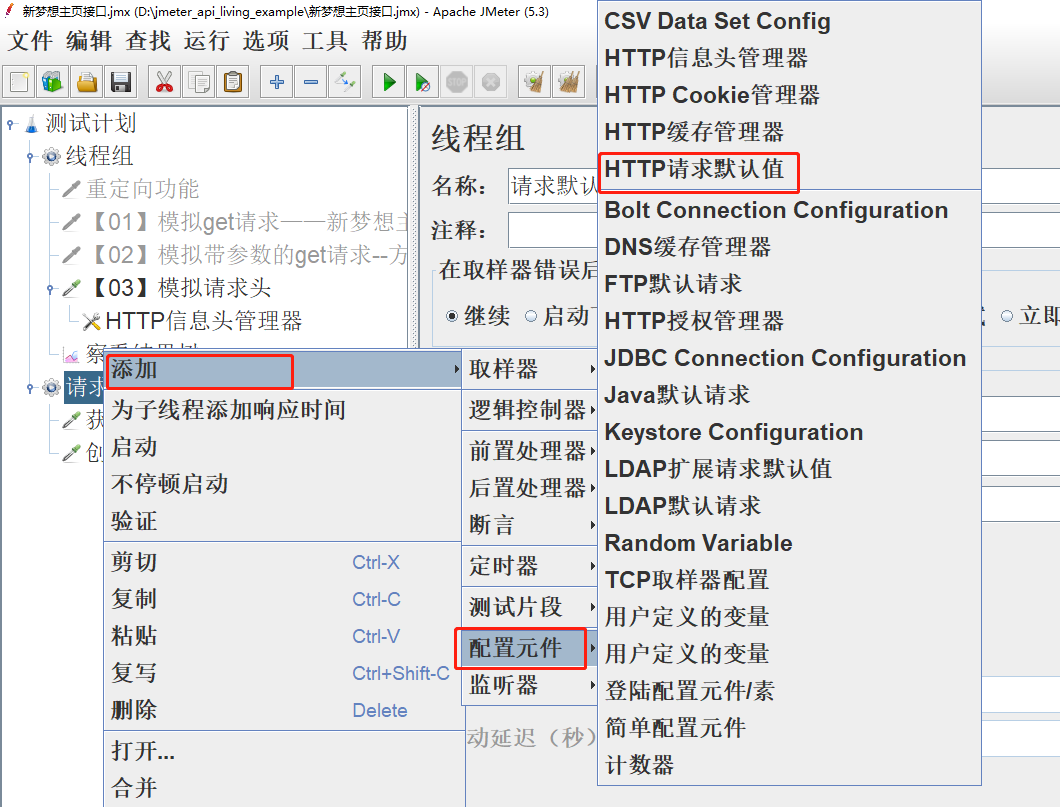


http信息头管理器添加方式：

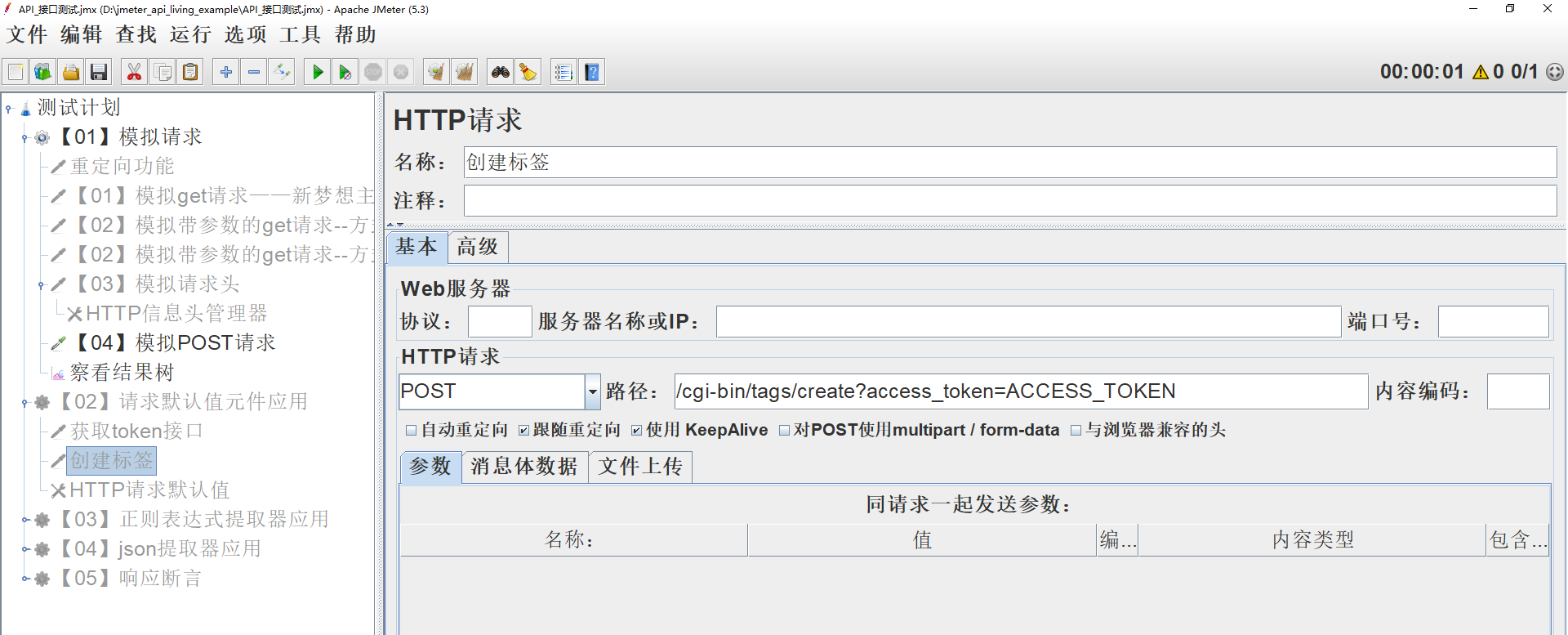


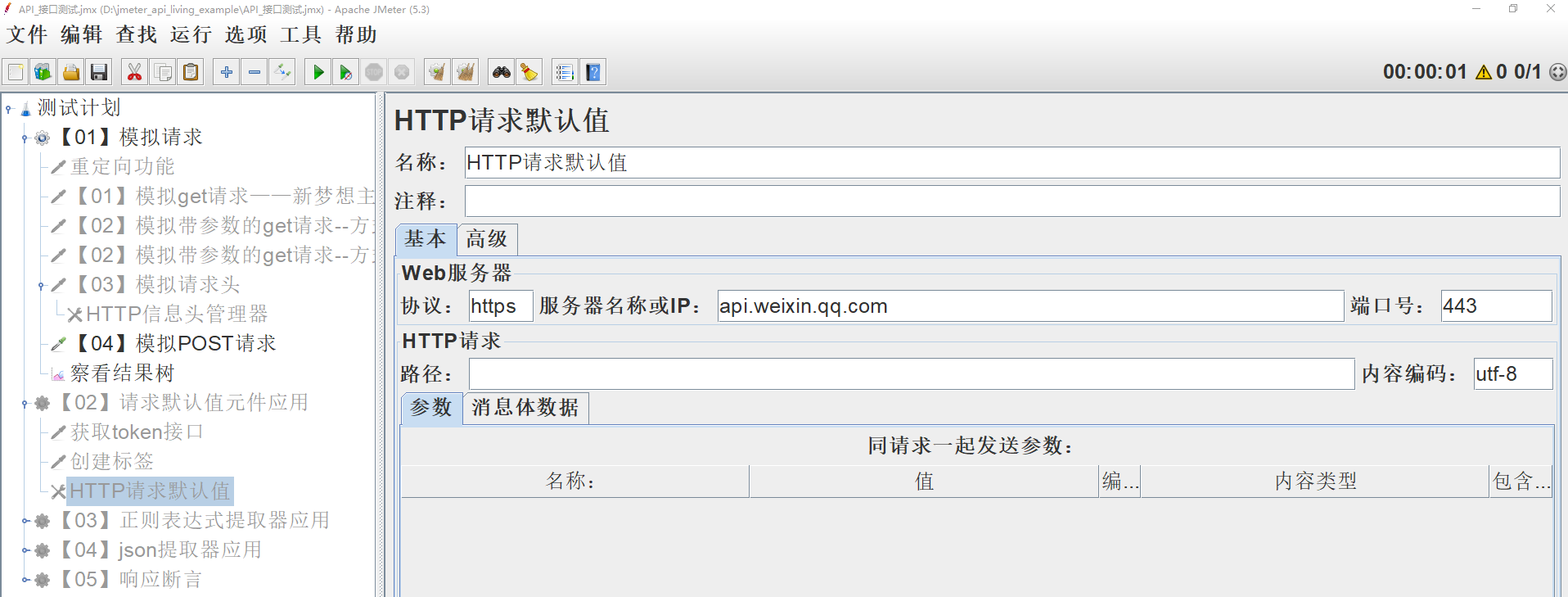
8、jmeter模拟请求默认值

http请求默认值添加方式：



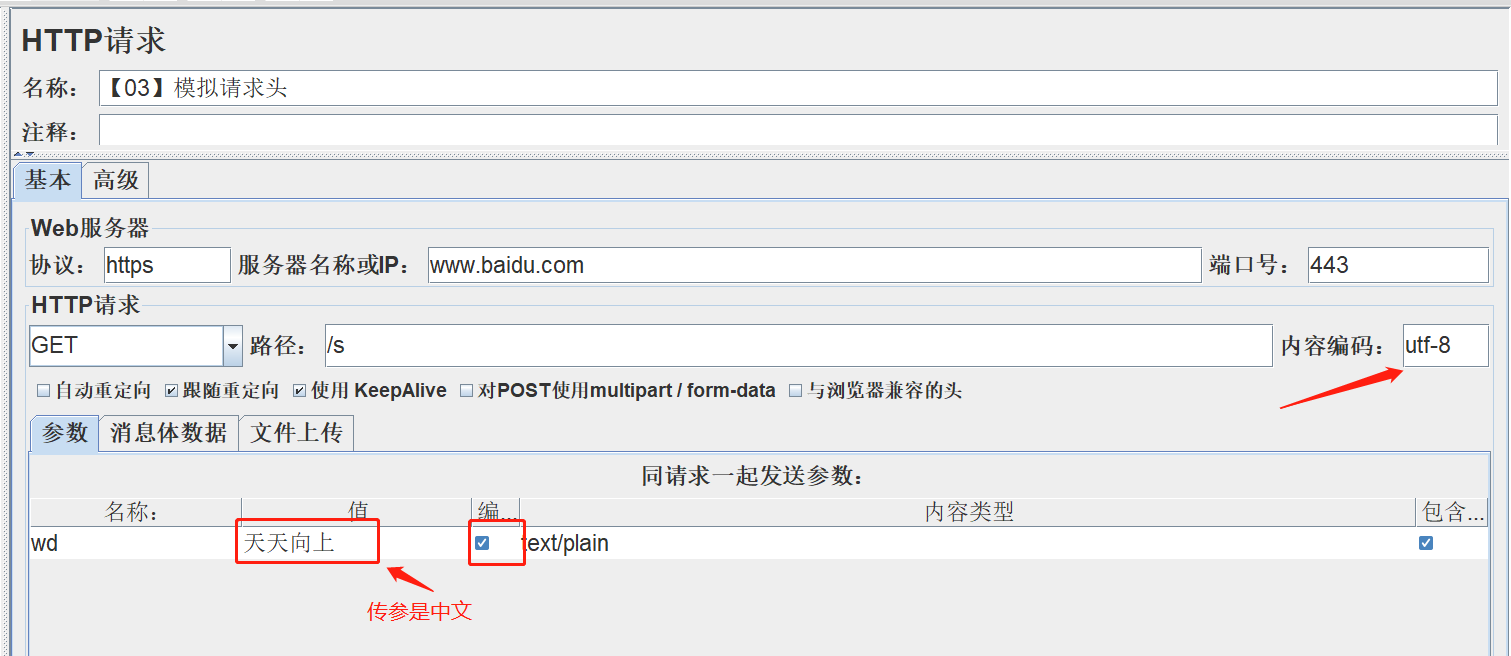






9、jmeter乱码处理

1) 请求参数乱码：



2) 解决查看结果树乱码：

进入jmeter的bin目录下，找到jmeter.properties文件，用UE等工具进行编辑，

找到sampleresult.default.encoding这个参数，将默认的ISO-8859-1修改成utf-8，

去掉注释符号，重启jmeter。



###### Jmeter接口测试（下）

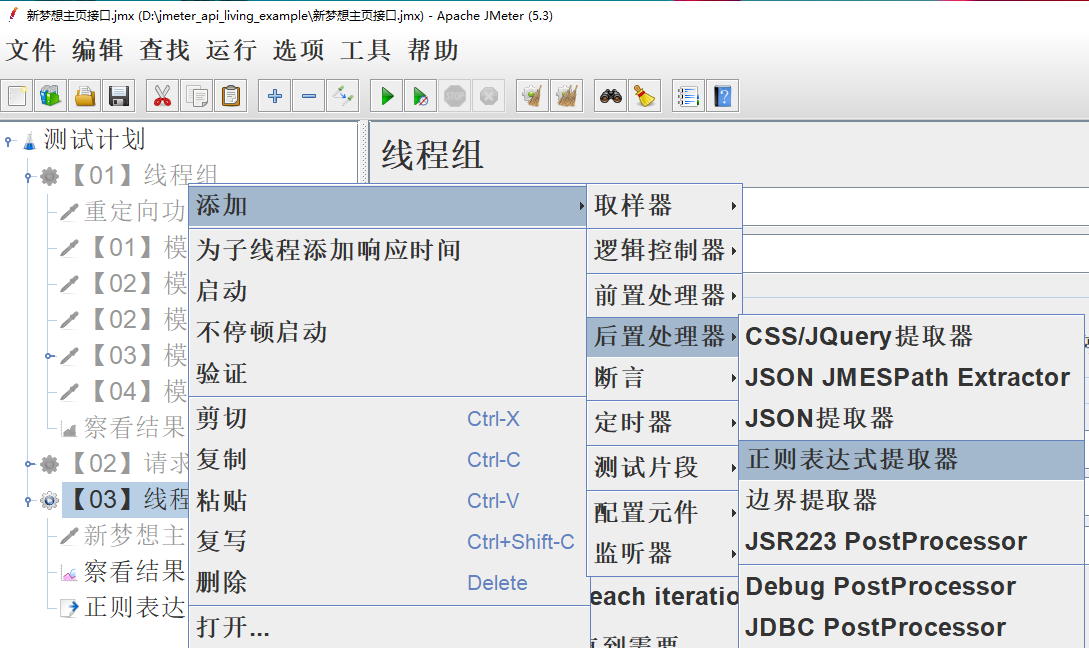
1. 正则表达式提取器元件

正则表达式提取器是位于后置处理器中的一个元件，它的作用是在请求发出并得到响应后，把响应信息中的部分通过正则表达式提取所需值，并将结果存储到给定的变量名中。

正则表达式截取的值是否准确，可以用调试取样器来查看。通过取样器--调试取样器添加该元件即可，默认查看变量的值，一般把调试取样器放置在所有取样器后面即可

在线正则表达式测试工具：工具网址：https://tool.oschina.net/regex

正则表达式提取器添加方式：



正则表达式提取器元件字段：

名称/注释：和之前讲的类似，忽略

APPly to:作用范围

Main sample and sub-samples:作用于主节点的取样器及对应子节点的取样器

Main sample only：仅作用于主节点的取样器

Sub-samples only:仅作用于子节点的取样器

JMeter Variable:作用于jmeter变量(输入框内可输入jmeter的变量名称)，

从指定变量值中提取需要的值

要检查的响应字段：

1）主体：响应报文的主体，最常用

2）Body(unescaped)：主体，是替换了所有的html转义符的响应主体内容，

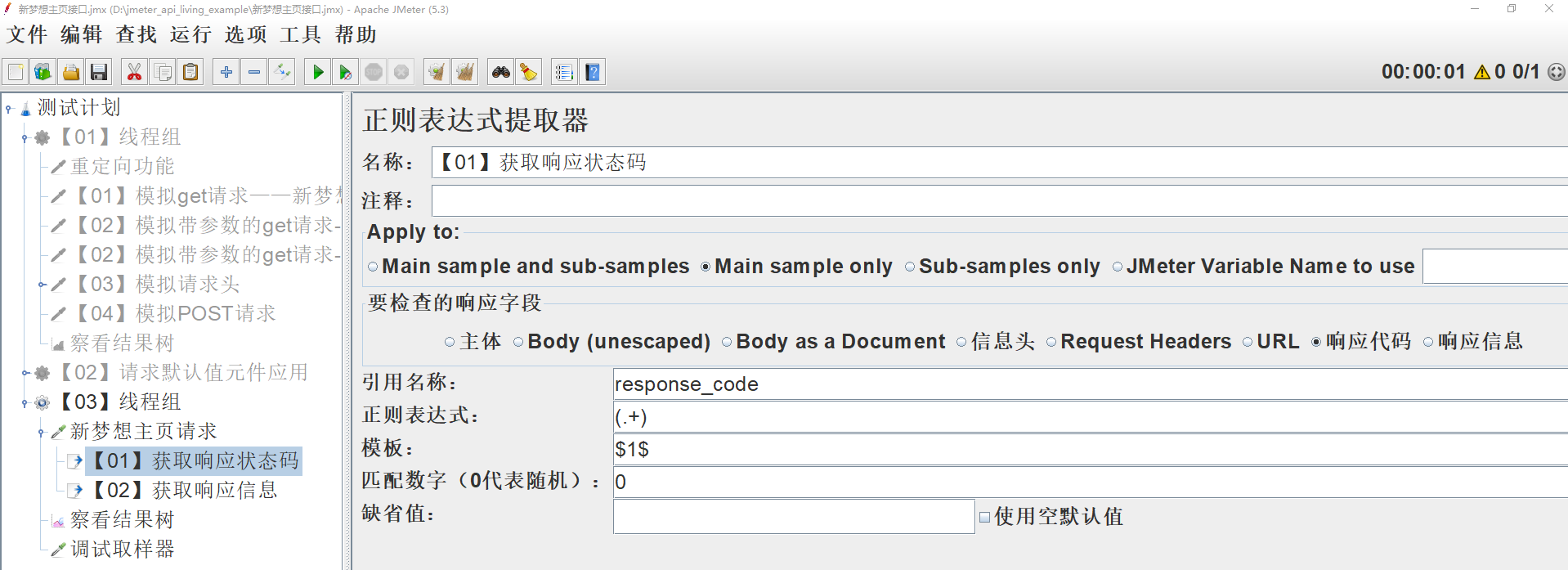
注意html转义符处理时不考虑上下文，因此可能有不正确的转换，不建议使用

3）Body as a Document：从不同类型的文件中提取文本，注意这个选项影响性能

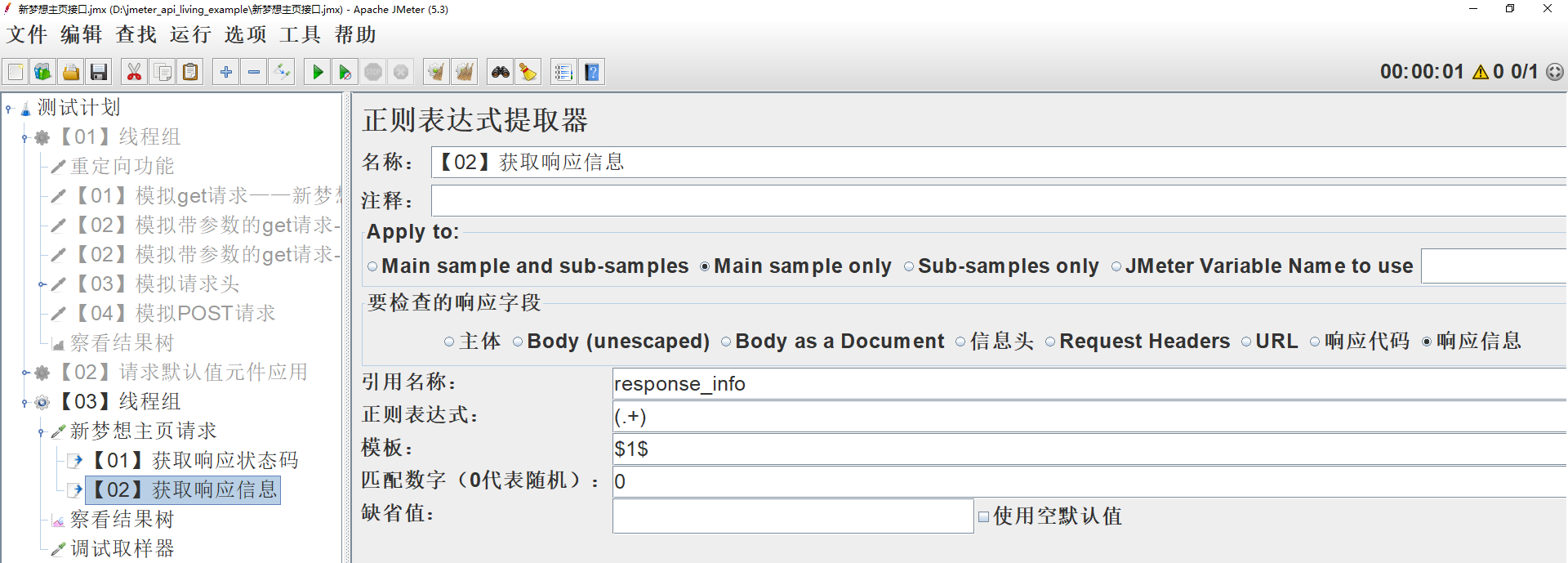
4）信息头：响应头部信息；Request Headers:请求信息头；URL：请求url；

响应代码：比如200、404等；响应信息：响应正文。

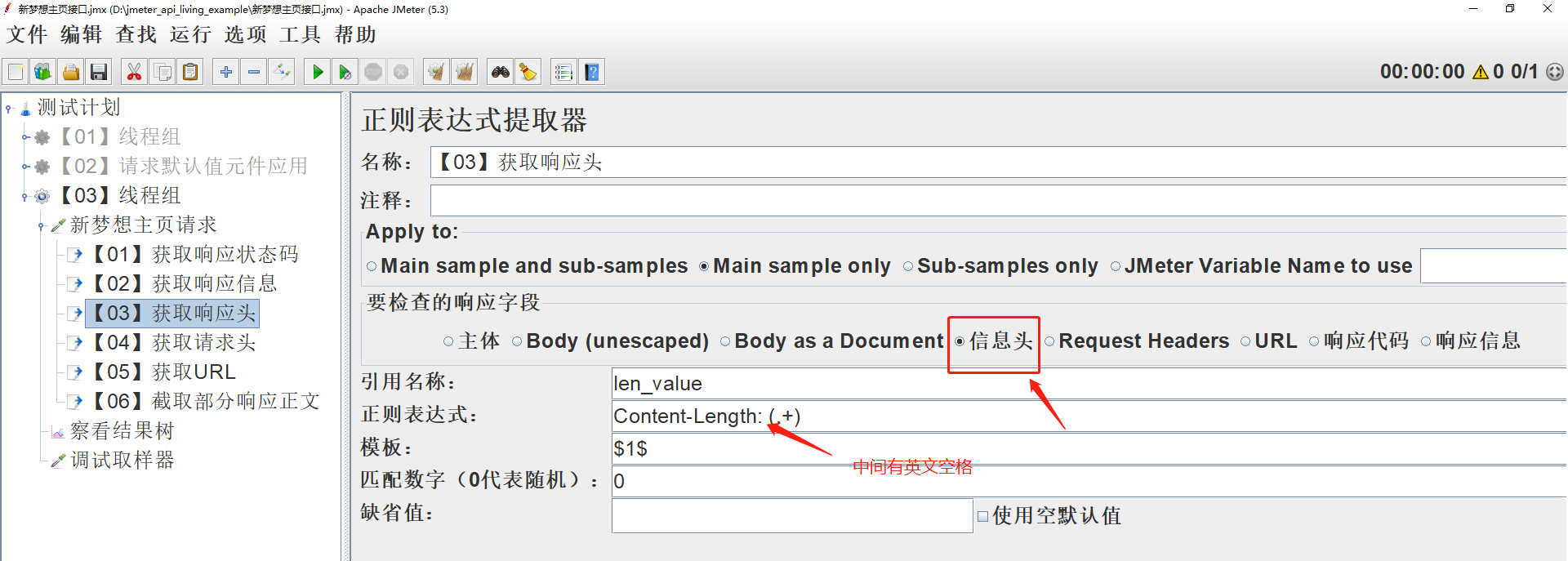
1. 获取响应状态码



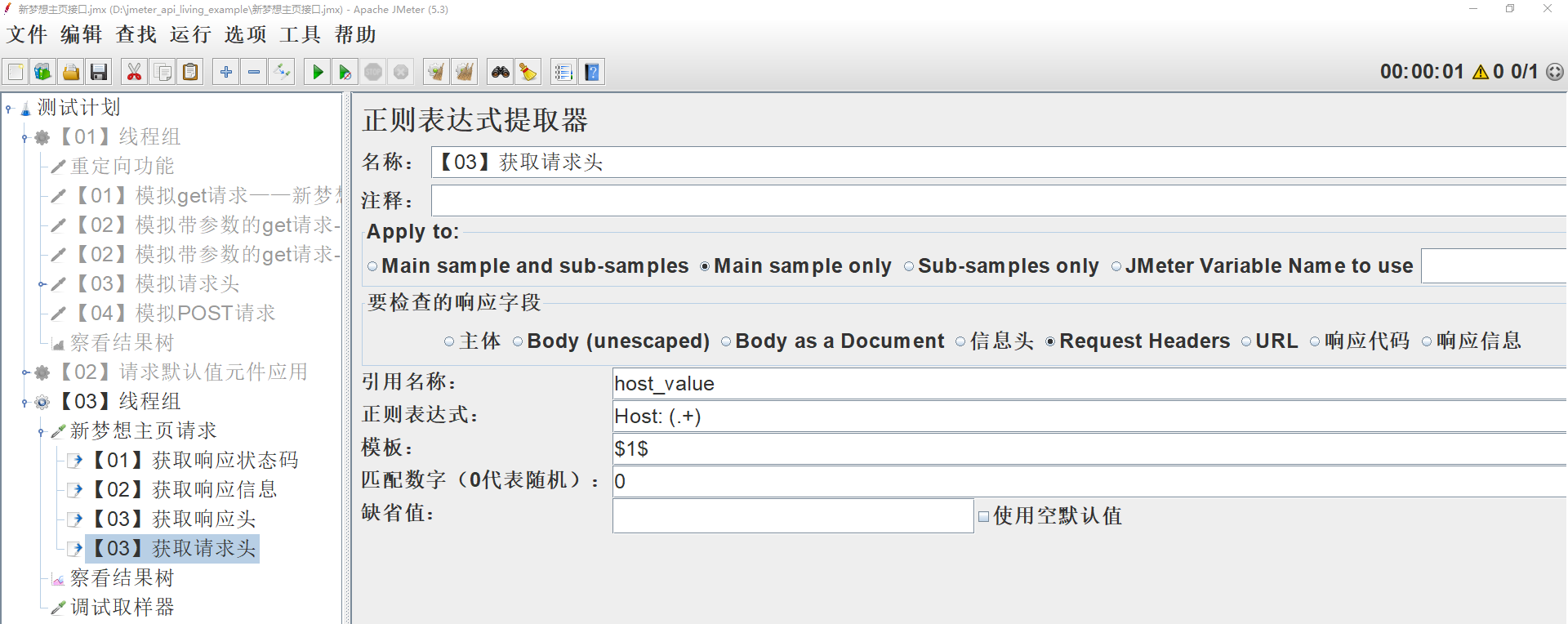
1. 获取响应信息



1. 获取响应头



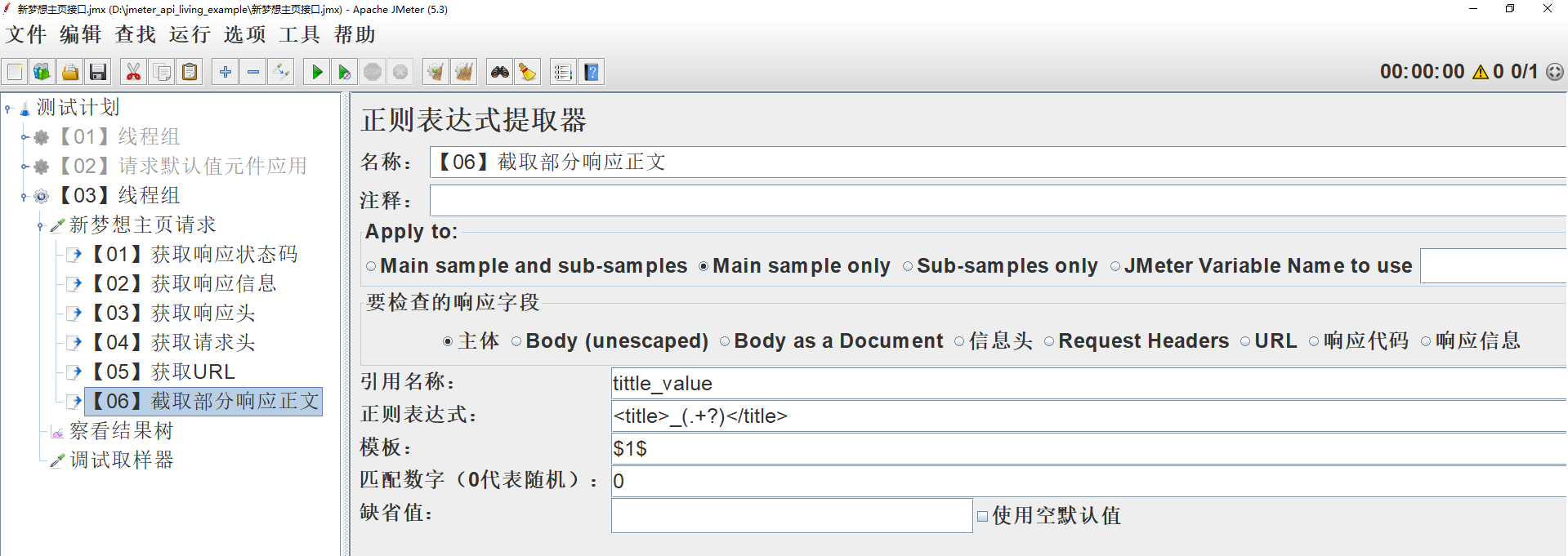
1. 获取请求头



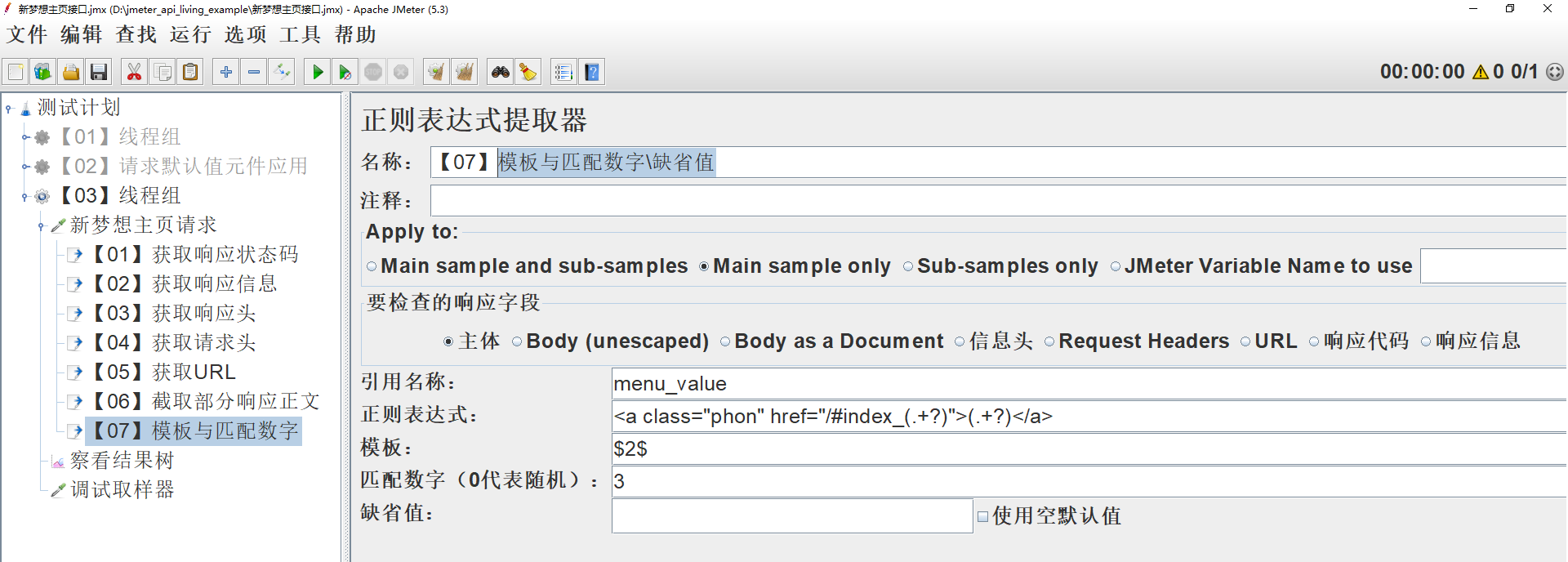
1. 获取URL



1. 截取部分响应正文



1. 模板与匹配数字\缺省值的使用



引用名称：Jmeter变量的名称，存储提取的结果；

如果在下个请求中想引用这个变量，语法：${引用名称}

正则表达式：使用正则表达式解析响应结果，“()”表示提取字符串中的部分值。

常用的正则表达式为 (.+?)

模板：表示提取到第几个值

$-1$:表示取所有匹配模板值

$0$:表示随机取匹配模板值

$1$:表示取第一个匹配模板值

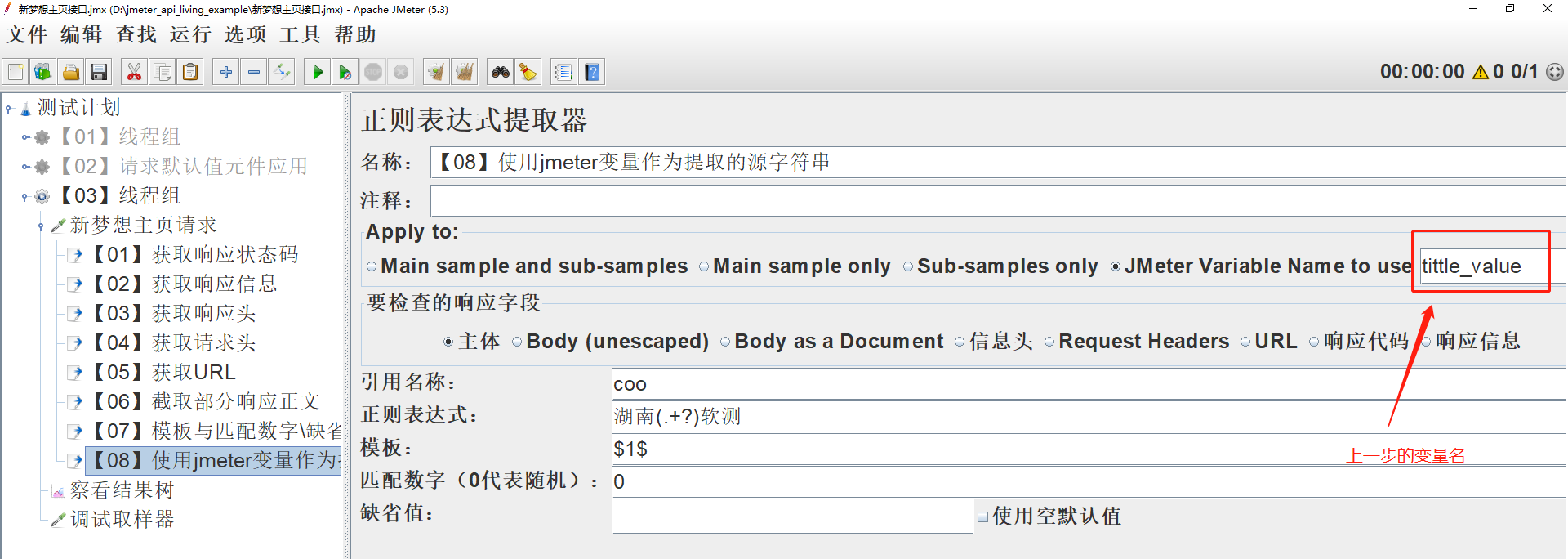
$2$:表示取第二个匹配模板值

$n$:n表示正整数，提取第n个匹配模板值

匹配数字：0便是随机取；-1表示取所有；正数表示对应取第一个匹配到的值

缺省值：缺省值相当于默认值；当未匹配到数据时，缺省值就是引用变量的值

1. 使用jmeter变量作为提取的源字符串

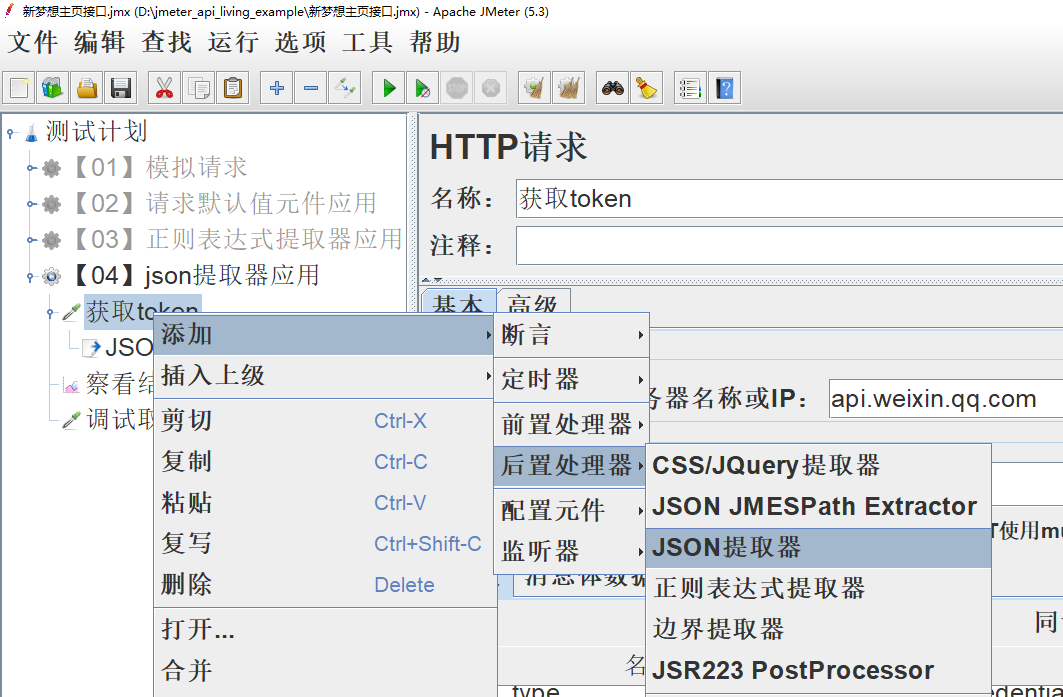


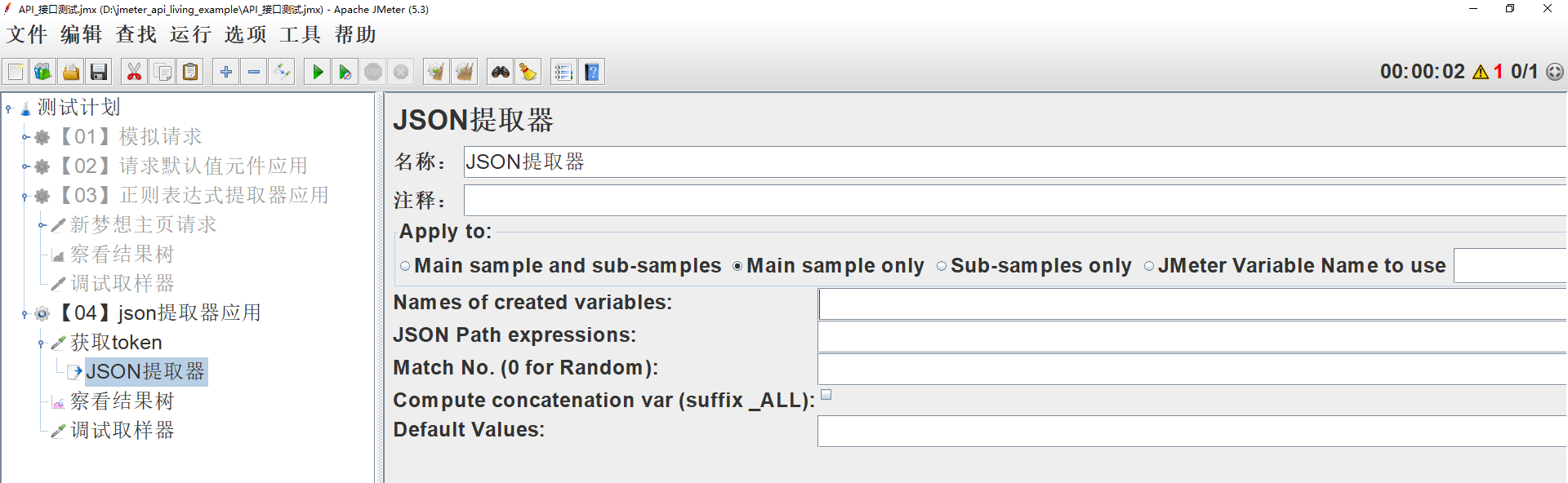
1. json提取器元件

json提取器与正则表达式提取器功能类似，也可以用来截取响应信息的部分保存到指定变

量中去，它只能用来处理响应正文，并且响应正文必须是json格式。

json提取器的添加方式：





json提取器元件字段：

APPly to: 作用范围

Names of created variables：jmeter变量的名称

JSON Path expressions：json表达式

Match No.(0 for Random)：匹配数字（0代表随机，1代表第一个，-1代表所有）

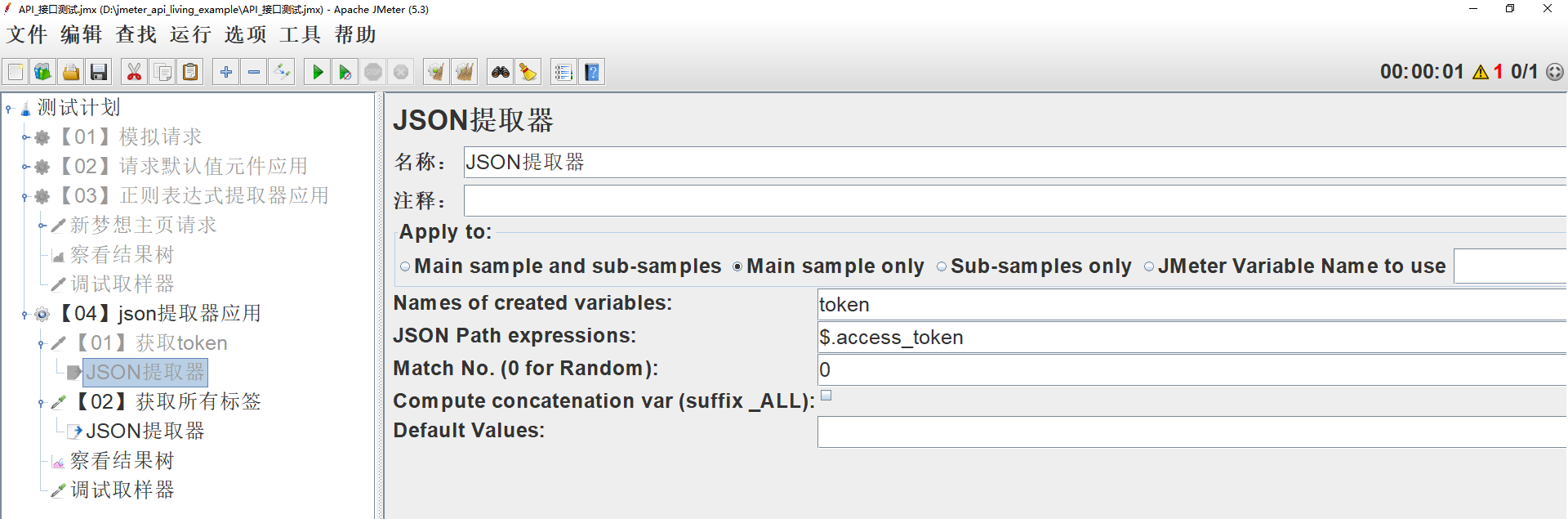
默认不填写是获取符合条件的第一个

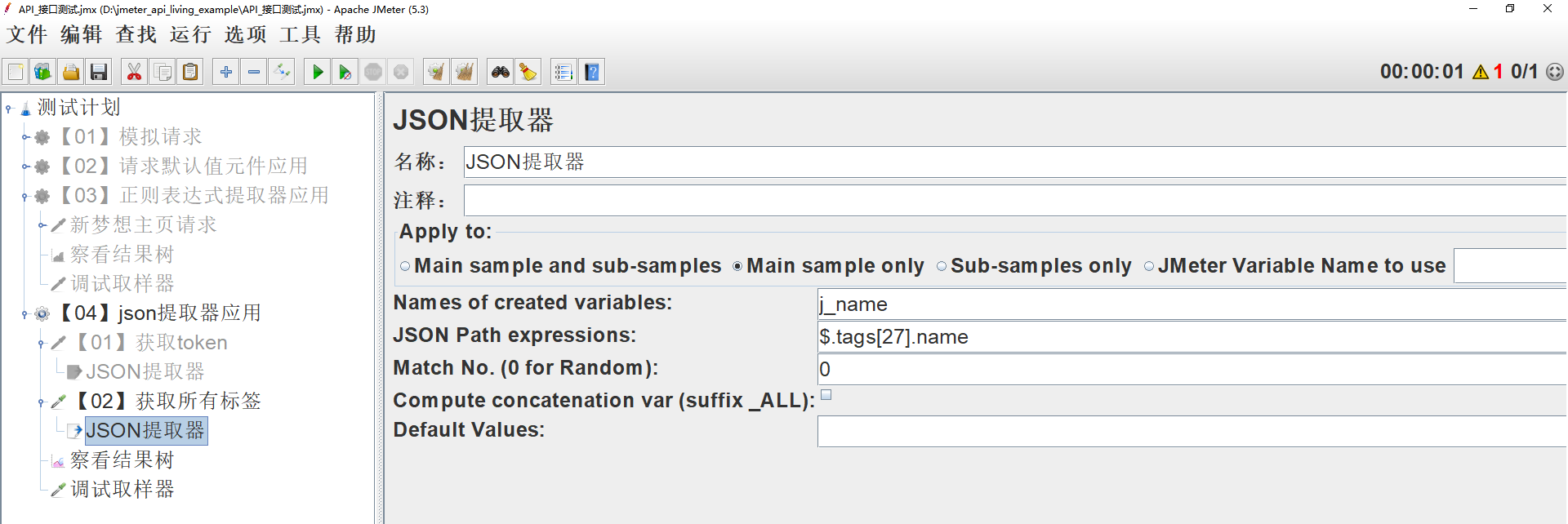
Compute concatenation var(suffix\_ALL)：如果发现许多结果，插件将使用“,”分隔符将它们连接起来，并将其存储在名为<variable name>\_ALL的var中

Default Values：当没有获取到参数值时的默认值

注意：当json提取器要提取两个值（或两个以上）的时候，一定要填写默认值，不然会报错

1. 获取响应正文

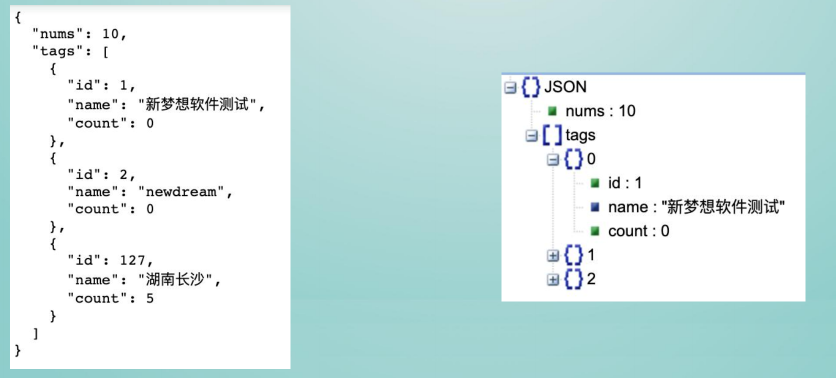




jsonpath语法：

$.nums ： 其中$代表根目录，这个表达式能取出10

$.tags[0].name: 遇到数组，使用[下标]



1. 响应断言元件

jmeter中该元件支持将请求或响应的各个字段与模式字符串进行比较

响应断言元件字段介绍：

名称/注释：和之前讲的类似

APPly to:作用范围 前面有描述

测试字段：要进行断言的响应字段选择，支持请求信息、响应信息等，

其中响应文本表示通过Apache Tika从各种类型的文档中提取文本（请参阅查看结果树文档视图部分），平时应用较少

忽略状态：取样器的总体成功取决于断言结果与现有的响应状态的结合，当选择 “忽略状态” 复选框时，响应状态被强制成功，再进行本次断言评估



a、模式匹配规则：

包括：如果文本包含正则表达式模式，则为true

匹配：如果整个文本与正则表达式匹配，则为true

相等：如果整个文本等于测试模式字符串，则为true（区分大小写）

字符串：如果文本包含测试模式字符串，则为true（区分大小写）

否：反转检查结果（先进行非运算，再进行与运算）

用于表示不包含、不匹配、不等于、没有和模式字符串相匹配的字符串

或者：一个断言可以添加多个“要测试的模式”，

使用OR选项后，只要其中有一个模式匹配，断言将会成功。

不选择“OR”时将默认为AND选项，必须所有模式都匹配，断言才会成功

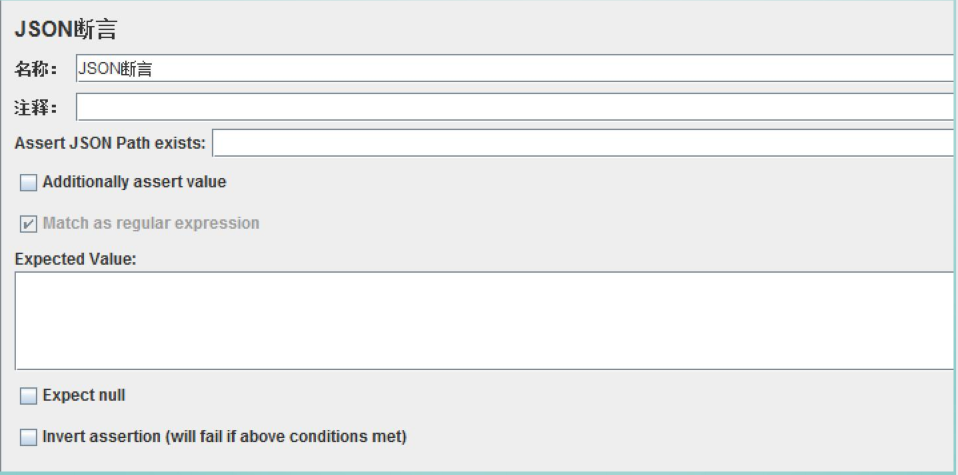
b、测试模式：输入结果期望值，可以增加多行期望结果

c、自定义失败消息：设置自定义的断言失败消息来替换系统默认的失败信息

d、忽略状态：jmeter对于响应代码为4\*\* 5\*\*自动红色报错处理

1. json断言元件

json断言元件只针对响应正文为json数据格式的内容进行断言功能



json断言元件字段介绍：

名称/注释：和之前讲的类似，忽略

Assert JSON Path exists：断言jsonpathpath表达式截取的值是否存在，这里输入要检查字段的jsonpath表达式即可

Addittionally assert value：是否添加断言值， 添加的断言值和json返回字段值进行比较，等于特定的值。如不勾选，只比对是否返回了该json字段

Match as regular expression：断言值是否支持正则表达式

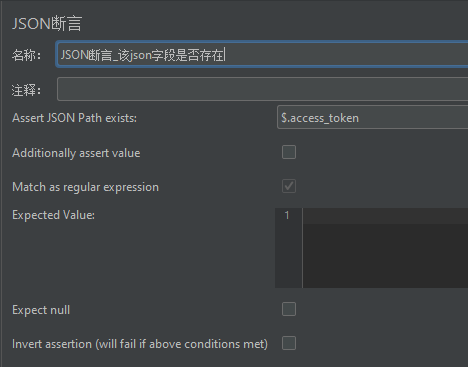
Expected Value：添加断言值

Expect null：预期空值，勾选这项，则会匹配断言结果为null的情况

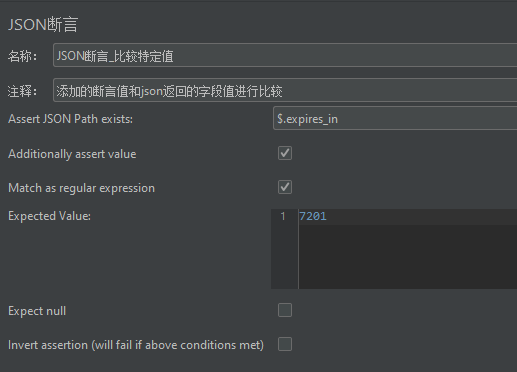
Invert assertion：取反，若上述断言结果为true，勾选此项，则断言结果为false ；

若上述断言结果为false，勾选此项，则断言结果为true

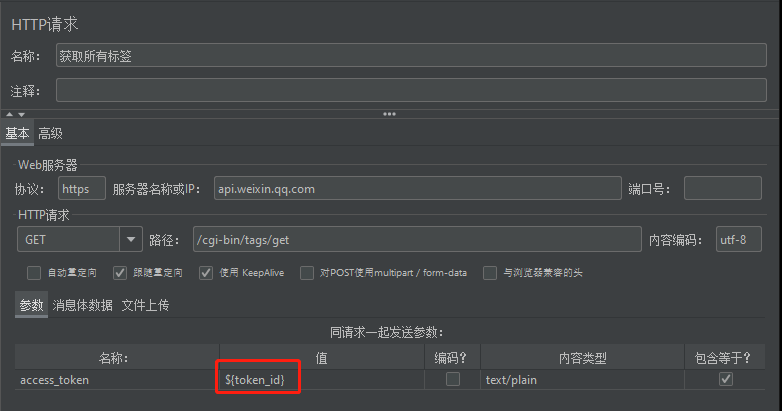
1. 断言该json字段是否存在



2）添加的断言值和json返回的字段值进行比较，等于特定的值



1. jmeter实现关联
2. 上一个接口中，设置json提取器或正则表达式提取器
3. 使用${变量名}把上一个接口截取的值进行引用



1. jmeter参数化实现

jmeter提供多种参数化设置的方式，常用的有：

1. 使用“用户定义的变量元件”实现

1. 使用“用户参数元件”实现
2. 使用“函数助手”实现
3. 使用“CSV 数据文件设置元件”实现
4. 其它方式（数据库数据、beanshell等）实现

* 使用 用户定义的变量元件 实现参数化：

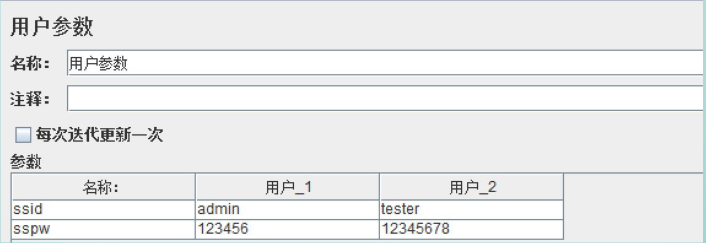
在实际使用jmeter进行接口测试过程中，我们把一些常用的配置值全部放置到用户定义的变量元件中，方便统一管理。在需要使用到的其他地方通过 ${变量名} 的语法格式进行引用即可。

“用户定义的变量元件”添加方式：配置元件->用户定义的变量



* 使用 用户参数元件 实现参数化：

用户参数元件可通过右键快捷菜单中选择 添加-前置处理器-用户参数进行添加，与用户自定义变量不同的是，用户参数列表中的参数取值可以在测试计划执行过程中发生变化。



每次迭代更新一次：如果选中该选项，则参数的值在每个迭代中保持不变，在新的迭代开始时取下一个值。如果取消该选项，则参数值在每个其作用域内的sampler发出请求时取下一个可用值。

建议在参数的取值范围很小，使用用户参数比较合适。

* 使用 函数助手 实现参数化：

函数助手中提供了很多可以进行参数化的函数，讲解常用典型的几个函数。

1. Random：随机整数函数，输入最小值、最大值，如果需要把生成的值保存在变量中，可以设置变量，如果只是一次使用，可以不设置存储变量。设置完上述三项后，点击生成即可生成函数代码，在需要的地方使用即可（使用方法${变量名}）。



1. RandomString：随机字符串函数，输入字符串指定长度、字符串的组成字符，

点击生成即可。



* 使用 CSV 数据文件设置元件 实现参数化：

csv数据文件中的csv称为逗号分隔值或者字符分割值,其文件以纯文本的形式存储表格数据。在jmeter中可以使用此类型文件提供的数据作为参数化。通过右击 -- 新增 -- 配置元件 -- CSV 数据文件设置元件进行添加。



界面字段介绍：

名称/注释：和之前讲的类似，忽略

文件名：指保存信息的文件目录，可以相对或者绝对路径

文件编码：默认为ANSI，根据文件类型来，一般设置为UTF-8

变量名称：给csv文件中各列起变量名字（有多列，用英文逗号隔开列名）

忽略首行：csv文件中如果是字段名，而不是实际的参数化数据的时候，

设置为true，表示第一行数据不作为参数化的数据

分隔符：用什么符号分隔参数化文件中的各列，以csv文件中的分割为准设置

是否允许带引号：允许参数化文件的参数带有双引号，默认是False，

如果参数文件的值带有双引号，False模式会将双引号也当做参数

遇到文件结束符再次循环：到了文件尾是否循环，True---继续从文件第一行开始读取，False---不再循环

遇到文件结束符停止线程：到了文件尾是否停止线程，True---停止，False---不停止，

注：当遇到文件结束符再次循环设置为True时，此项设置无效。

线程共享模式：设置csv数据文件的使用范围

1. jmeter参数化之取序列数据

a、利用函数助手对话框实现

在jmeter菜单处点击 工具 -- 函数助手对话框 -- 下拉框选择 counter -- 进入如下界面：

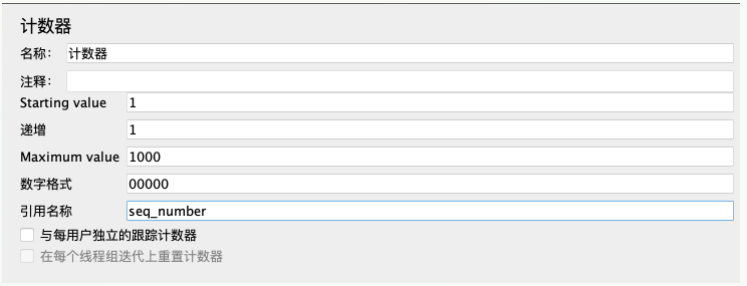


点击生成时会自动复制生成的函数，直接可以在需要的地方粘贴。如：



设置后，可以通过线程组设置界面的循环次数输入比如 5 来进行测试。

b、利用配置元件--计数器元件实现



界面字段介绍：

starting value : 初始值，计数器的初始值设置

递增：递增 为每次递增的值设置，

如starting value设置为1，递增设置为3 ，则会产生 1、4、7、10等的序列值

maximum value : 最大值设置

数字格式：设置序列的显示格式，

如设置为 00000，则会产生 00001 、00002、00003这样格式的值

引用名称：即设置参数名称，其它地方可以通过该参数名称进行引用

与每用户独立的跟踪计数器：全局的计数器，如果不勾选，即全局的，

比如用户#1 获取值为1，用户#2获取值还是为1；

如果勾选，即独立的，则每个用户有自己的值：

比如用户#1 获取值为1，用户#2获取值为2。

在每个线程组迭代上重置计数器：可选，仅勾选与每用户独立的跟踪计数器时可用；

如果勾选，则每次线程组迭代，都会重置计数器的值，

当线程组是在一个循环控制器内时比较有用。

如上图设置好后，在需要的地方如下图的方式引用：



8、元件的执行顺序&作用域

元件的执行顺序：

配置元件-->前置处理器-->定时器-->取样器-->后置处理器（除非Sampler 得到的返回结果为空）-->断言（除非Sampler 得到的返回结果为空）-->监听器（除非Sampler 得到的返回结果为空）

关于执行顺序，有两点需要注意：

1）前置处理器、后置处理器和断言等元件功能对取样器作用，因此，如果在它们的作用域内没有任何取样器，则不会被执行。

2）如果在同一作用域范围内有多个同一类型的元件，则这些元件按照它们在测试计划中的上下顺序依次执行。

元件的作用域：

配置元件：元件会影响其作用范围内的所有元件

前置处理器：元件在其作用范围内的每一个sampler元件之前执行

定时器：元件对其作用范围内的每一个sampler 有效

后置处理器：元件在其作用范围内的每一个sampler元件之后执行

断言：元件对其作用范围内的每一个sampler 元件执行后的结果执行校验

监听器：元件收集其作用范围的每一个sampler元件的信息并呈现

总结：从各个元件的层次结构判断每个元件的作用域

命令行运行jmeter脚本：jmeter -n -t “.jmx脚本路径” -l “测试结果.jtl”

生成的jtl文件可通过jmeter界面进行查看

9、基于jmeter开展企业接口自动化测试实战

在企业使用jmeter开展实际的接口自动化测试，建议按如下操作流程，可以使整个接口测试过程更规范，更有效。流程步骤如下：

1. 根据接口设计文档编写接口测试用例并评审
2. 把文档形式的接口测试用例录入到jmeter工具中
3. 在相应的测试阶段执行接口用例，得出结果报告

把接口测试用例录入到jmeter中详细操作：

1. 在测试计划下新增一个线程组，并对测试计划和线程组根据项目重新命名
2. 在线程组下新增一个用户定义的变量元件，进行URL等公共参数设置
3. 在线程组下新增一个全局的http请求默认值元件进行取样器公共部分的设置
4. 在线程组下根据公司接口测试需要新增事务控制器元件作为按模块组织接口的文件

夹（事务控制器元件类似文件夹的含义）

1. 在事务控制器下继续新增事务控制器作为单个接口的测试用例汇总文件夹
2. 在单个接口测试用例汇总文件夹中继续新增事务控制器作为单个测试用例的文件
3. 在单个用例文件下新增取样器和断言操作（一般断言操作可设置在最后一个请求中）
4. 反复根据2.5--2.7的过程增添接口测试用例
5. 添加察看结果树进行结果察看

