Jmeter:

构成Jmeter脚本的最小单位是元器件，不同复杂程度的脚本，主要体现在使用了不同类型的元器件，元器件之间的相互关联，相互依赖。本文意在热身，因此介绍最基本的用法。

**最简单的脚本**

左侧菜单树右击添加元器件：线程组，Http请求，查看结果树，聚合报告。

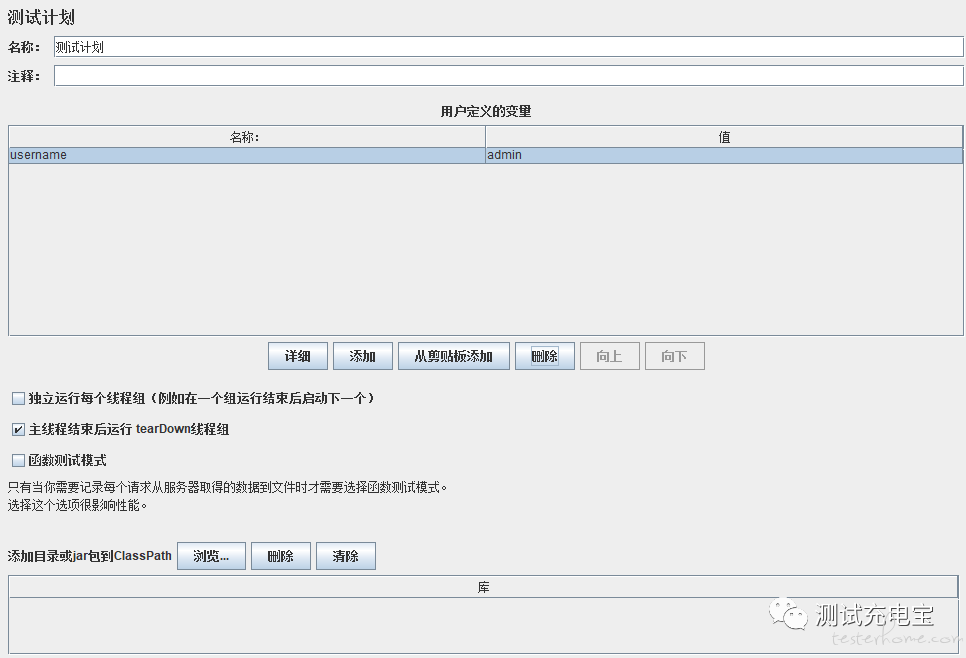
## 元器件分类

元器件有以下几大类：

取样器  
逻辑控制器  
前置处理器  
后置处理器  
断言  
定时器  
测试片段  
配置原件  
监听器  
每个分类下有很多元器件，加起来差不多有几十种，下面会对工作中最常用的元器件作详细说明。

**元器件介绍**

**测试计划**

[[](https://testerhome.com/uploads/photo/2020/06e4b7ab-d1aa-4341-aba4-a10cd2413e4c.png!large)](https://testerhome.com/uploads/photo/2020/06e4b7ab-d1aa-4341-aba4-a10cd2413e4c.png!large)

一般来说，测试计划不需要做任何配置，但还是选取部分配置做个说明

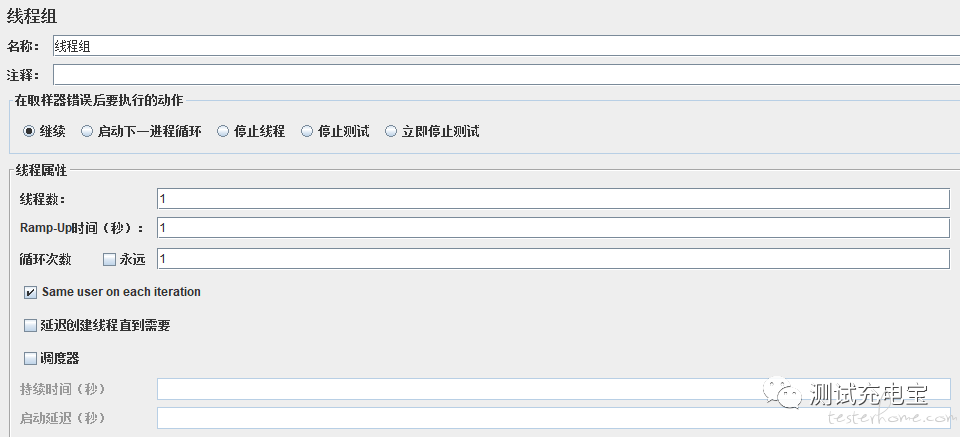
**用户自定义变量**

设置全局变量，key-value的形式存在，调用方式${username}，

**添加目录或jar包到classpath**

引用外部jar包，可以在此处添加，保证可以使用jar中方法

## 线程组

[[](https://testerhome.com/uploads/photo/2020/7bbf8940-8302-4785-acb1-7712076d9ec1.png!large)](https://testerhome.com/uploads/photo/2020/7bbf8940-8302-4785-acb1-7712076d9ec1.png!large)

**在取样器错误后要执行的动作**

* 继续：忽略错误，继续执行
* 启动下一个进程循环：忽略错误，线程当前循环终止，执行下一个循环。
* 停止线程：当前线程停止执行，不影响其他线程正常执行。
* 停止测试：整个测试会在所有当前正在执行的线程执行完毕后停止
* 立即停止测试：整个测试会立即停止执行，正在执行的取样器会被中断。

**线程属性**

* 线程数：也就是并发数，每个线程将会完全独立的运行测试计划，互不干扰。多个线程用于模仿对服务器的并发访问
* Ramp-Up时间(秒)：启动所有线程所需要的时间，例如5个线程，5秒启动时间，相当于1秒启动1个线程
* 循环次数：每个线程循环的次数，如果设置为1，那么只执行测试计划一次

**调度器**

* 持续时间（秒）：控制测试执行的持续时间
* 启动延时（秒）：控制测试在多久后启动执行

## 总结

本章节主要介绍了九类元件，并且对线程组和测试计划的配置项做了详细说明。

# 三、Jmeter 基础-常用元器件介绍 (2)

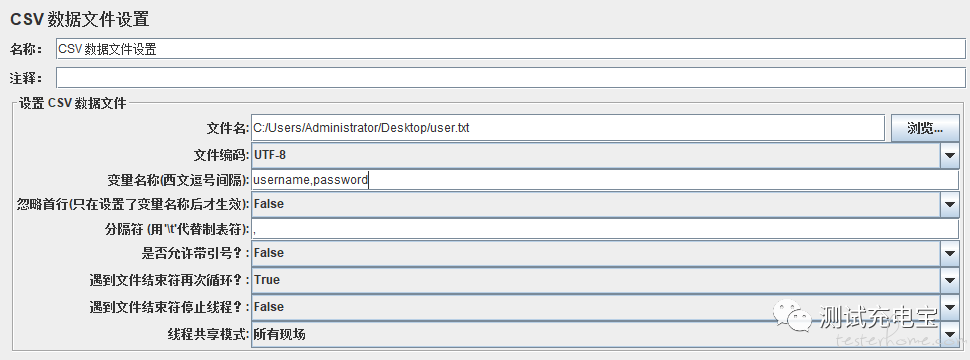
[zhusp](https://testerhome.com/zhusp) · 4 天前 · 最后由 [gaodq](https://testerhome.com/romantico21) 回复于 3 天前 · 419 次阅读

 目录

## 配置原件

重要元器件：CSV数据文件设置，HTTP信息头管理器，HTTP Cookie管理器，HTTP默认请求值，用户自定义变量

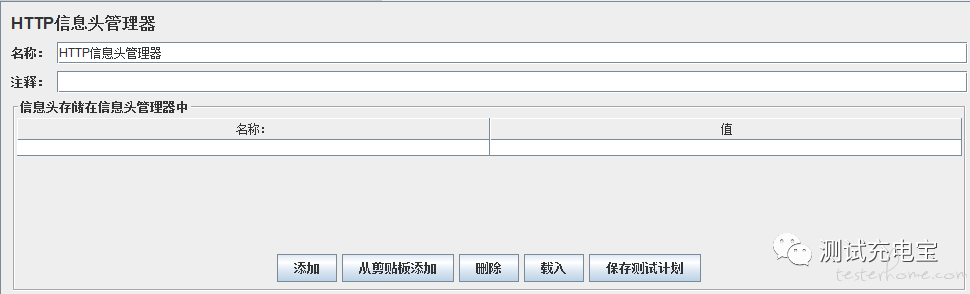
## CSV 数据文件设置

[[](https://testerhome.com/uploads/photo/2020/69526489-b6d3-4ccd-974b-580df43e75ed.png!large)](https://testerhome.com/uploads/photo/2020/69526489-b6d3-4ccd-974b-580df43e75ed.png!large)

**设置csv数据文件**

* 文件名：你的csv数据文件路径,还支持txt,bat等
* 文件编码：例如可以是UTF-8，那么jmeter就会以UTF-8读取文件
* 变量名称：以逗号分隔的名称列表，它与CSV文件中的列相匹配
* 忽略首行：如果第一行是列名而不是测试数据，就选择True忽略掉列名
* 分隔符：默认为','也可以设置为其他值（建议使用默认）
* 是否允许带引号：默认为False，如果测试数据中带有引号则设置为Ture
* 遇到文件结束符再次循环：默认为True，意思读完最后一行在回来读取第一行
* 遇到文件结束符停止线程：读完最后一行停止线程
* 线程共享模式：定义如何在并发线程之间分配值

## HTTP信息头管理器

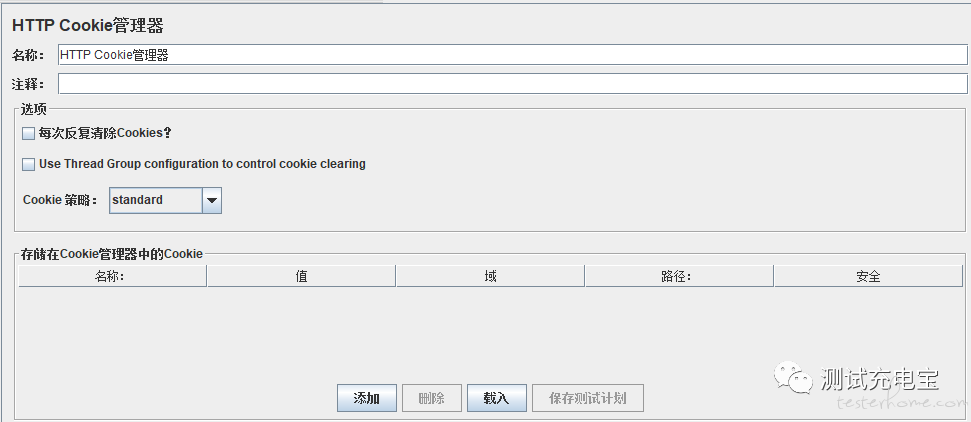
[[](https://testerhome.com/uploads/photo/2020/84172c33-562f-4d29-9f8a-ad74a0df6b65.png!large)](https://testerhome.com/uploads/photo/2020/84172c33-562f-4d29-9f8a-ad74a0df6b65.png!large)

请求页面或接口时附加的协议头（调试脚本报错，问题可能出在请求头上），常用的信息头如下（我在工作中经常用到的）

## 协议 说明

1 Accept 可接受的响应内容类型  
2 Host 表示服务器的域名以及服务器所监听的端口号  
3 Content-Type 请求体的MIME类型  
4 Referer 表示浏览器所访问的前一个页面  
5 User-Agent 浏览器的身份标识字符串  
6 Connection 客户端（浏览器）想要优先使用的连接类型  
7 Accept-Encoding 可接受的响应内容的编码方式

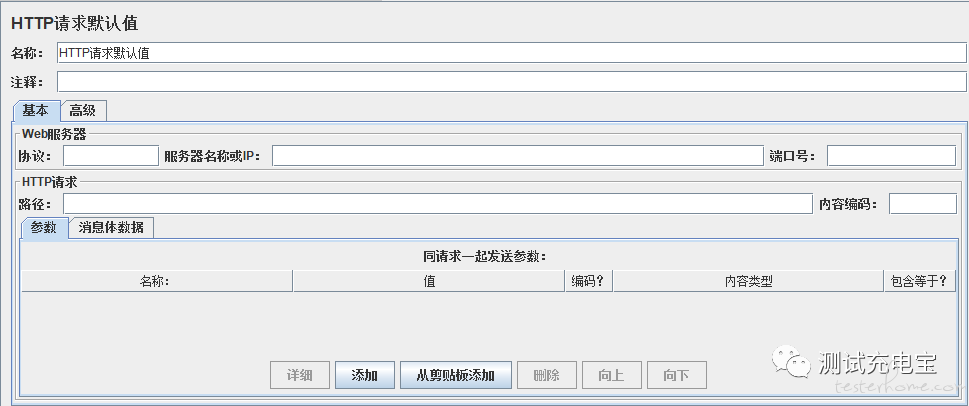
## HTTP Cookie管理器

[[](https://testerhome.com/uploads/photo/2020/ce93d6c0-96eb-4e5d-a09c-b0ba5a59024a.png!large)](https://testerhome.com/uploads/photo/2020/ce93d6c0-96eb-4e5d-a09c-b0ba5a59024a.png!large)

在做登录后的页面请求，登录成功后HTTP Cookie管理器会写入cookies，保证后面的请求是登录状态

注意：如果在一个测试计划内有多个Cookie Manager ，Jmeter目前无法指定哪个被使用。所以，一个测试计划内最好只有一个cookie manager。并且，一个manager里的 cookie 并不能被其它manager所引用。所以在使用多个Cookie Managers 时要谨慎。

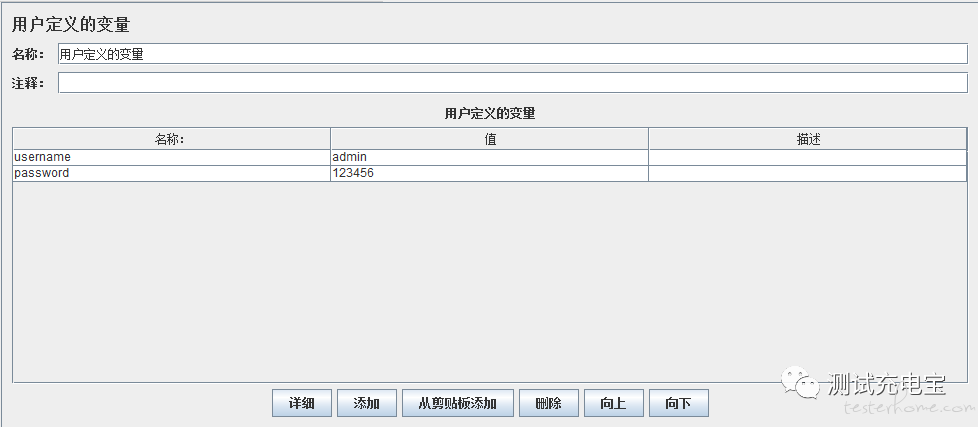
## HTTP请求默认值

[[](https://testerhome.com/uploads/photo/2020/56d10967-62dd-4004-811e-fabdec60bc62.png!large)](https://testerhome.com/uploads/photo/2020/56d10967-62dd-4004-811e-fabdec60bc62.png!large)

使用HTTP请求默认值，作用域内的HTTP请求无需设置IP和端口，只需设置路径，运行时自动把HTTP请求默认值和HTTP请求进行拼接。

好处在于如果想切换环境，只需更改HTTP请求默认值。

## 用户自定义变量

[[](https://testerhome.com/uploads/photo/2020/8983b51e-c63d-435f-9366-47099ff0881a.png!large)](https://testerhome.com/uploads/photo/2020/8983b51e-c63d-435f-9366-47099ff0881a.png!large)

变量管理器，放在线程组下相当于全局变量，调用变量方式：${变量名}

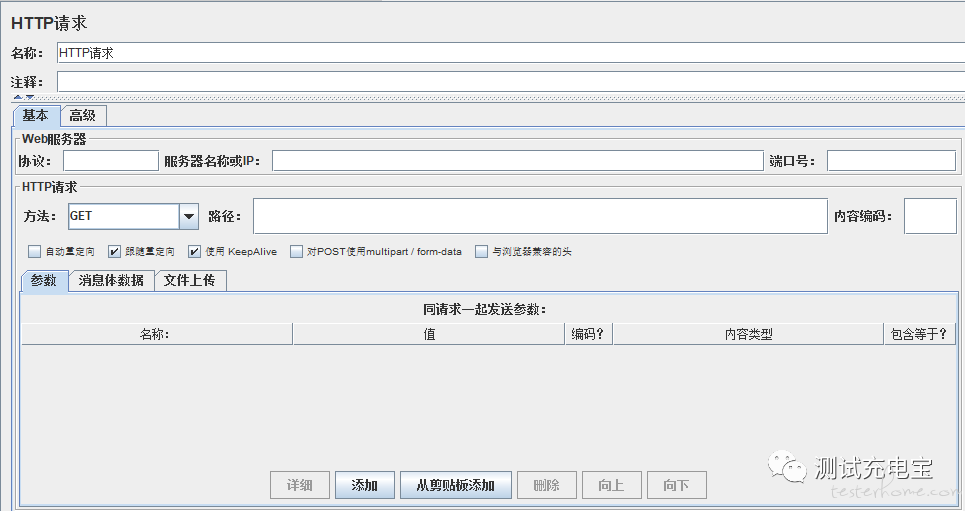
名称：变量名

值：变量的值

## 取样器

重要元器件：HTTP请求，JDBC Connection Configuration，JDBC Request

## HTTP请求

[[](https://testerhome.com/uploads/photo/2020/00ec7806-fc23-4116-bfce-4eaae49ae173.png!large)](https://testerhome.com/uploads/photo/2020/00ec7806-fc23-4116-bfce-4eaae49ae173.png!large)

脚本中运用最多的元器件，使用也比较简单，按照抓包或者接口文档对号入座即可

协议：http或https

服务器名称或IP：服务器域名或者IP

端口：如果服务器名称是域名可省略，是IP则那就查看下具体端口号

方法：GET,POST,PUT,DELETE等

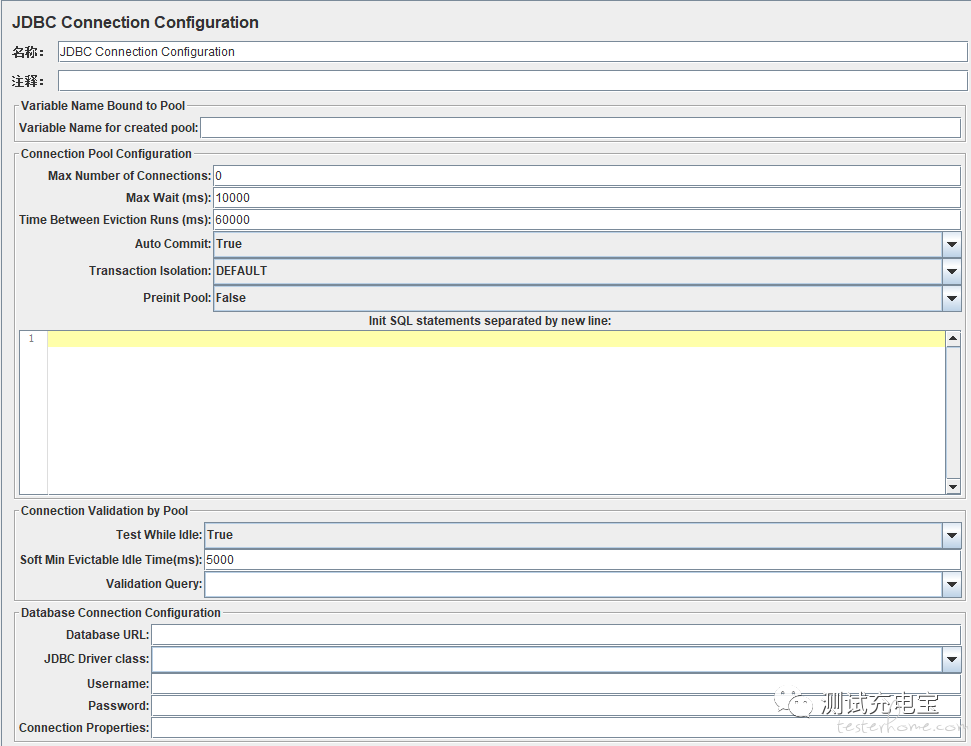
路径：接口或页面路径，例如/login

参数：key-value形式

消息体数据：json数据提交在此设置

文件上传：上传文件才用，一般用的不多

## JDBC Connection Configuration和JDBC Request

[[](https://testerhome.com/uploads/photo/2020/82a01612-9a5b-4ab3-9918-fe8171ed6b92.png!large)](https://testerhome.com/uploads/photo/2020/82a01612-9a5b-4ab3-9918-fe8171ed6b92.png!large)

Variable Name Bound to Pool

Variable Name：变量名绑定到池的唯一标识。与JDBC request中的相对应，决定JDBC取样的配置。简单理解就是在JDBC request的时候确定去哪个绑定的配置

Connection Pool Configuration（建议默认）

Max Number of Connections：数据库连接池的最大连接数，一般可设置为0，意思是每个线程都使用单独的数据库连接，线程之间数据库连接不共享

Max Wait (ms)：在连接池中取回连接最大等待时间

Time Between Eviction Runs (ms)：数据库空闲清理的间隔时间

Auto Commit：自动提交sql语句

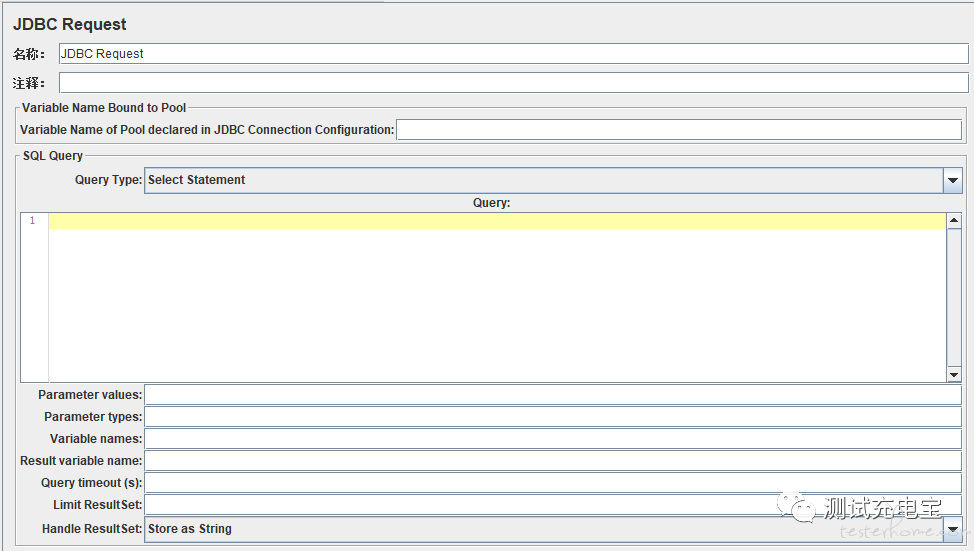
Database Connection Configuration

Database URL: jdbc:mysql://服务器地址:3306/数据库名

JDBC Driver class:数据库JDBC驱动类名：com.mysql.jdbc.Driver

Username:数据库连接用户名

password:数据库连接密码

[[](https://testerhome.com/uploads/photo/2020/63d08e30-4426-4b9b-833d-70f4ff6fda19.png!large)](https://testerhome.com/uploads/photo/2020/63d08e30-4426-4b9b-833d-70f4ff6fda19.png!large)

Variable Name Bound to Pool

Variable Name ....：数据库连接池的名字，需要与JDBC Connection Configuration的Variable Name Bound Pool名字保持一致

SQL Query

Query Type：此处支持方式多样，可以用于添加或者筛选数据，根据需要和Query配合使用

Query：填写的sql语句未尾不要加“;”

Parameter valus：参数值，顺序替代Query中的?；

Parameter types：参数类型

Variable names：保存sql语句返回结果的变量名 ，用于作为参数供调用

Result variable name：创建一个对象变量，保存所有返回的结果 ，供调用

Query timeout：查询超时时间

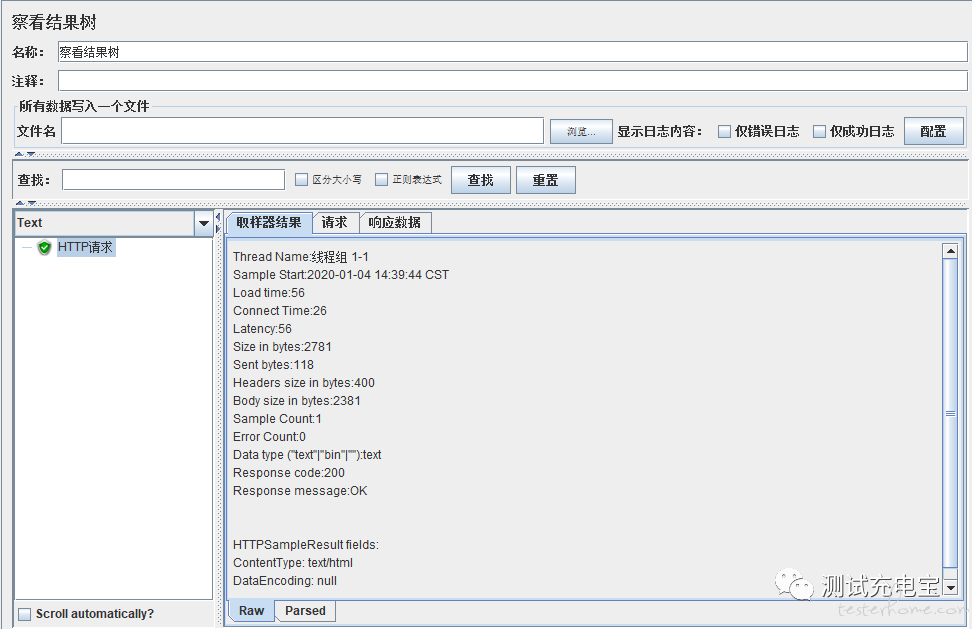
Handle result set：定义如何处理由callable statements语句返回的结果

JDBC Request取样器和JDBC Connection Configuration配置原件一起使用，前者主要用于连接数据库，前者用于执行sql语句保存变量供调用

## 监听器

重要元器件：查看结果树，聚合报告

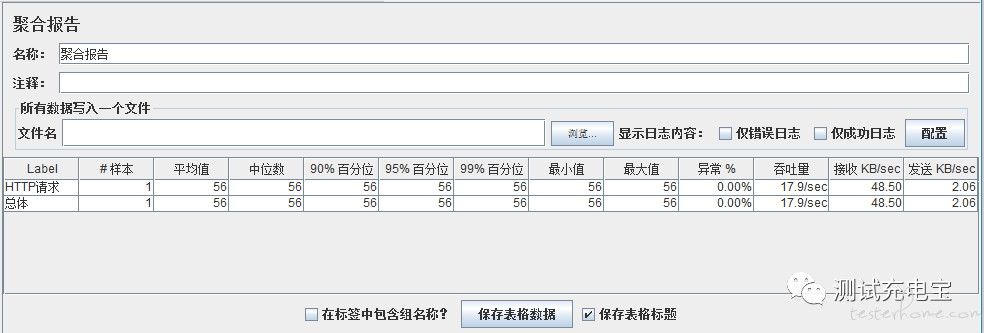
## 查看结果树

[[](https://testerhome.com/uploads/photo/2020/0f1750c7-0879-4720-9990-c84b8e91c567.png!large)](https://testerhome.com/uploads/photo/2020/0f1750c7-0879-4720-9990-c84b8e91c567.png!large)

左侧展示执行过的所有的取样器，成功显示绿色否则显示红色（没加断言绿色不一定代表成功）

右侧展示请求和响应的明细，有点类似浏览器F12Network下的详细信息或者Fiddler抓包详情

聚合报告

[[](https://testerhome.com/uploads/photo/2020/0fc5cb6b-a83f-4d1a-8828-1db1d9bf8f73.png!large)](https://testerhome.com/uploads/photo/2020/0fc5cb6b-a83f-4d1a-8828-1db1d9bf8f73.png!large)

统计每一个请求的请求数，平均值，最大，最小值，错误率，大约吞吐量(以请求数/秒为单位)和以kb/秒为单位的吞吐量

Label:请求的名称，就是我们在进行测试的httprequest sampler的名称

#样本:总共发给服务器的请求数量，如果模拟10个用户，每个用户迭代10次，那么总的请求数为：10\*10 =100次；

平均值:默认情况下是单个 Request 的平均响应时间，当使用了 Transaction Controller 时，以Transaction 为单位显示平均响应时间 ，单位是毫秒

中位数: 50%用户的请求的响应时间，中位数

90%百分位:90%的请求的响应时间

95%百分位:95%的请求的响应时间

99%百分位:99%的请求的响应时间

最小值:最小的响应时间

最大值:最大的响应时间

异常%:错误率=错误的请求的数量/请求的总数

吞吐量: 默认情况下表示每秒完成的请求数（Request per Second），当使用了 Transaction Controller 时，也可以表示类似 LoadRunner 的 Transaction per Second 数

接收KB/sec: 每秒从服务器端接收到的数据量

发送KB/sec: 每秒从服务器端接收到的数据量

特别说明：90% 百分位参数正确的含义

一组数由小到大进行排列，找到他的第90%个数（假如是3），那么这个数组中有90%的数将小于等于3 。

用在性能测试的响应时间也将非常有意义，也就是90%请求响应时间不会超过3 秒。

## 总结

本章节介绍的元器件比较多，我经常用到的是篇幅中介绍的监听器和配置元件下所有，以及HTTP请求。

# 四、Jmeter 基础-常用元器件介绍 (3)

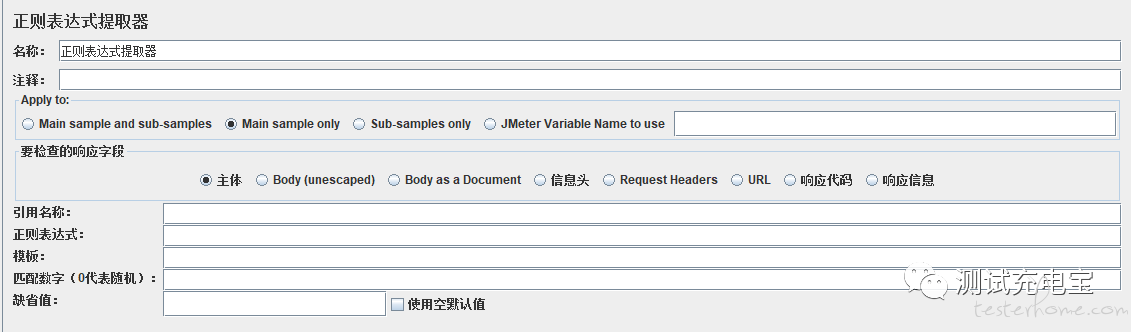
## 前置处理器

前置处理器中，我觉得BeanShell 预处理程序，用户参数，JDBC 预处理程序用的比较多  
JDBC 预处理程序：跟JDBC request类似，不做过多介绍  
BeanShell 预处理程序：准备在进阶篇中介绍，基础篇不做过多介绍，会用到java  
用户参数：我在工作中用的很少，还是喜欢用户自定义变量

## 后置处理器

主要元器件：正则表达式提取器，JSON提取器，边界提取器，XPath提取器，BeanShell 后置处理程序

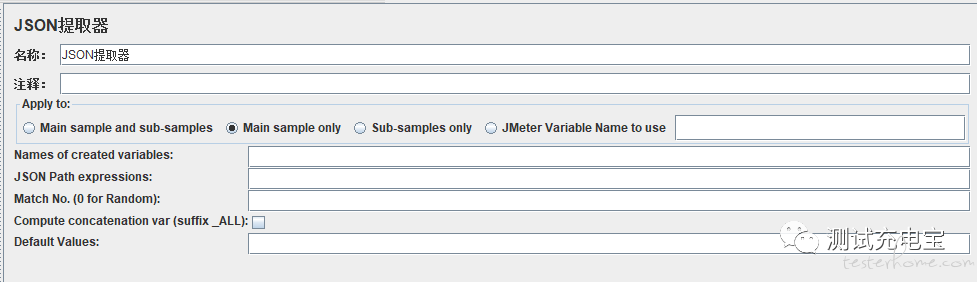
## 正则表达式提取器

[[](https://testerhome.com/uploads/photo/2020/6db8d5bf-84fb-4e46-9cbf-df26070c13e7.png!large)](https://testerhome.com/uploads/photo/2020/6db8d5bf-84fb-4e46-9cbf-df26070c13e7.png!large)

**APPly to**  
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**  
主体：响应报文的主体，最常用  
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，不太建议使用  
Body as a Document：从不同类型的文件中提取文本，注意这个选项比较影响性能  
信息头：响应信息头（如果你使用的是中文版的Jmeter，会看到这一项是信息头，这是中文翻译问题，应以英文为准）  
Request Headers:请求信息头  
URL：请求url  
响应代码:比如200、404等  
响应信息:响应信息  
**参数设置说明**  
引用名称：参数名称，可以被调用  
正则表达式：使用正则表达式解析响应结果，正则的基本使用方法可参考正则表达式的官方说明  
模板：如果正则表达式有多个提取结果，则结果是数组形式，模板$1$，$2$等等，表示把解析到的第几个值赋给变量；从1开始匹配，以此类推。若只有一个结果，则只能是$1$；  
匹配数字（0代表随机）：正则表达式匹配数据的结果可以看做一个数组，表示如何取值：0代表随机取值，正数n则表示取第n个值（比如1代表取第一个值），负数则表示提取所有符合条件的值。  
缺省值：默认值，取不到的情况，参数名称赋的值  
**举例**  
view=1&token=721652512141&name=ggui  
要取token值  
正则表达式：token=(\d+)&name=ggui

特别说明  
".\*?"在正则中非常好用，能解决大部分匹配  
上面表达式可改写成token=(.\*？)&name=ggui

## JSON提取器

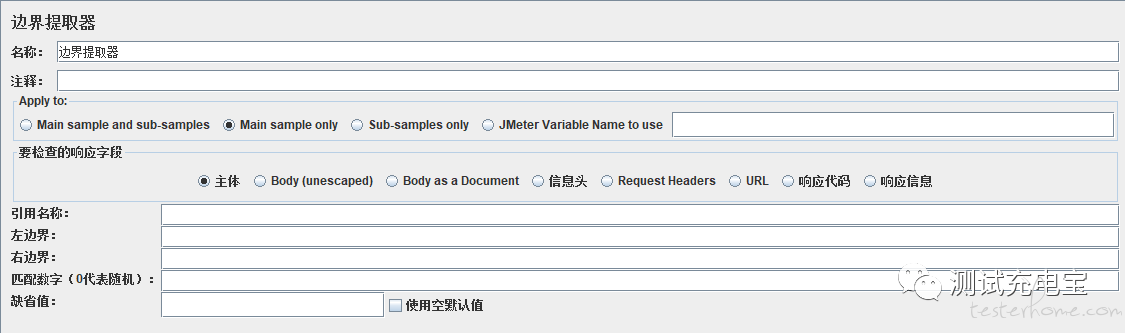
[[](https://testerhome.com/uploads/photo/2020/2a7b45d3-4442-4497-bcd8-b01ff5d7af3d.png!large)](https://testerhome.com/uploads/photo/2020/2a7b45d3-4442-4497-bcd8-b01ff5d7af3d.png!large)

用于提取请求返回结果中的某个值或者某一组值，标准写法为$.key，其中key为返回结果中的一个键，如果是多层则继续用.key即可，如果遇到key的value值为一个List，则使用.key[n]，其中n为list中元素的索引  
**参数设置说明**  
Names of created variables:参数名称，可以被调用  
JSON Path expressions:表达式  
Match No.(0 for Random)：匹配编号，-1匹配所属有，0随机，其他则从1开始  
Default Values：默认值，取不到的情况，参数名称赋的值  
举例

{  
 "data":[  
 {  
 "name":"张三",  
 "age":18  
 },  
 {  
 "name":"李四",  
 "age":28  
 }  
 ]  
}

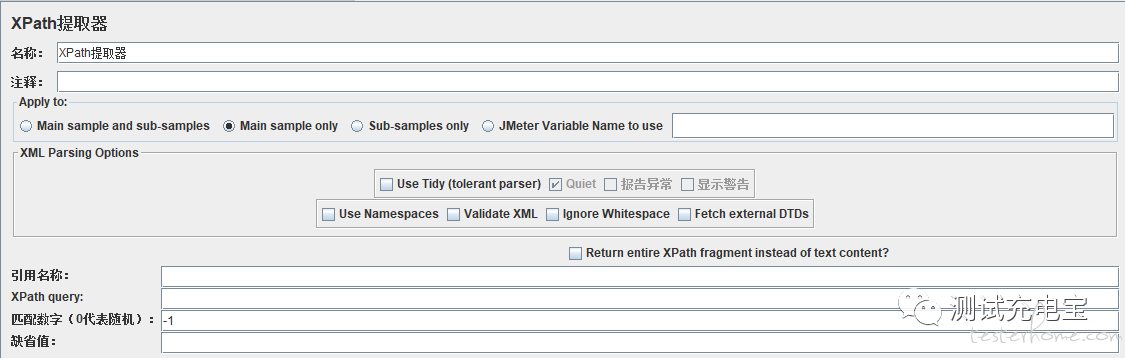
**要取第一个人的姓名**  
JSON Path expressions：$.data[0].name  
**要取年龄>18的姓名**  
JSON Path expressions：$.data[?([*@*.age](https://testerhome.com/.age)>18)].name

## 边界提取器

[[](https://testerhome.com/uploads/photo/2020/2c833df1-96a8-425b-bc44-915d15537a89.png!large)](https://testerhome.com/uploads/photo/2020/2c833df1-96a8-425b-bc44-915d15537a89.png!large)

**参数设置说明**  
引用名称：参数名称，可以被调用  
左边界：要提取字符的左边  
有边界：要提取字符的右边  
匹配数字（0代表随机）：匹配编号，-1匹配所属有，0随机，其他则从1开始  
缺省值：默认值，取不到的情况，参数名称赋的值  
**举例**  
view=1&token=721652512141&name=ggui  
左边界：token=  
右边界：&name  
可得到结果721652512141  
**PS：注意，左右边界尽可能唯一，能更精确匹配到想提取的字符**

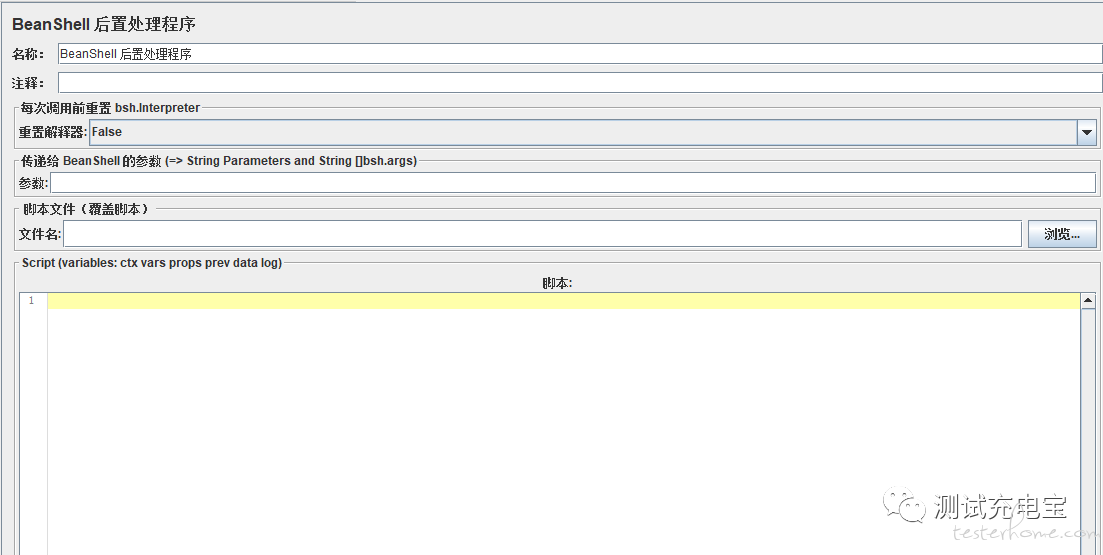
## XPath提取器

[[](https://testerhome.com/uploads/photo/2020/60e27bf0-10bc-4ac0-98db-2fedda9a385f.png!large)](https://testerhome.com/uploads/photo/2020/60e27bf0-10bc-4ac0-98db-2fedda9a385f.png!large)

**XML Parsing Options（要解析的XML参数）**  
Use Tidy（tolerant parser）：当需要处理的页面是HTML格式时，必须选中该选项；如果是XML或XHTML格式（例如RSS返回），则取消选中  
Quiet：表示只显示需要的HTML页面  
报告异常：表示显示响应报错  
显示警告：表示显示警告  
Use Namespaces：如果启用该选项，后续的XML解析器将使用命名空间来分辨  
Validate XML：根据页面元素模式进行检查解析  
Ignore Whitespace：忽略空白内容  
Fetch external DTDs：如果选中该项，外部将使用DTD规则来获取页面内容  
Return entire XPath fragment of text content：返回文本内容的整个XPath片段  
**参数设置说明**  
引用名称：参数名称，可以被调用  
XPath Query：用于提取值的XPath表达式  
匹配数字：匹配编号，-1匹配所属有，0随机，其他则从1开始  
缺省值：默认值，取不到的情况，参数名称赋的值  
**举例**

**要获取【百度一下】按钮上的文字**  
XPath Query：//input[[*@*id](https://testerhome.com/id)="su"]/@value

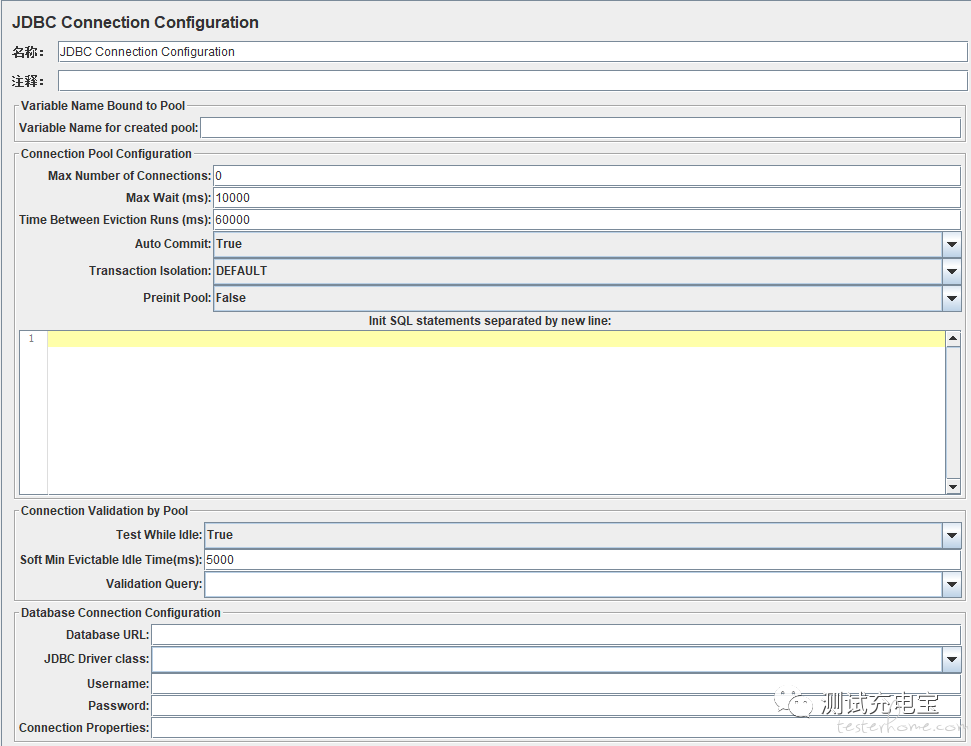
## BeanShell 后置处理程序

[[](https://testerhome.com/uploads/photo/2020/cb352c4a-d16d-4f31-871f-a78a4e0feeb2.png!large)](https://testerhome.com/uploads/photo/2020/cb352c4a-d16d-4f31-871f-a78a4e0feeb2.png!large)

介于本阶段是基础篇，Beanshell相关的放在进阶篇介绍，这里不做过多讲解

## 总结

本章节讲述前置和后置处理器，前置处理器的元器件和Beanshell后置处理器会在进阶篇介绍，后置处理器介绍了最常用的正则表达式提取器，JSON提取器，边界提取器，XPath提取器。

[[](https://testerhome.com/uploads/photo/2020/82a01612-9a5b-4ab3-9918-fe8171ed6b92.png!large)](https://testerhome.com/uploads/photo/2020/82a01612-9a5b-4ab3-9918-fe8171ed6b92.png!large)

Variable Name Bound to Pool

Variable Name：变量名绑定到池的唯一标识。与JDBC request中的相对应，决定JDBC取样的配置。简单理解就是在JDBC request的时候确定去哪个绑定的配置

Connection Pool Configuration（建议默认）

Max Number of Connections：数据库连接池的最大连接数，一般可设置为0，意思是每个线程都使用单独的数据库连接，线程之间数据库连接不共享

Max Wait (ms)：在连接池中取回连接最大等待时间

Time Between Eviction Runs (ms)：数据库空闲清理的间隔时间

Auto Commit：自动提交sql语句

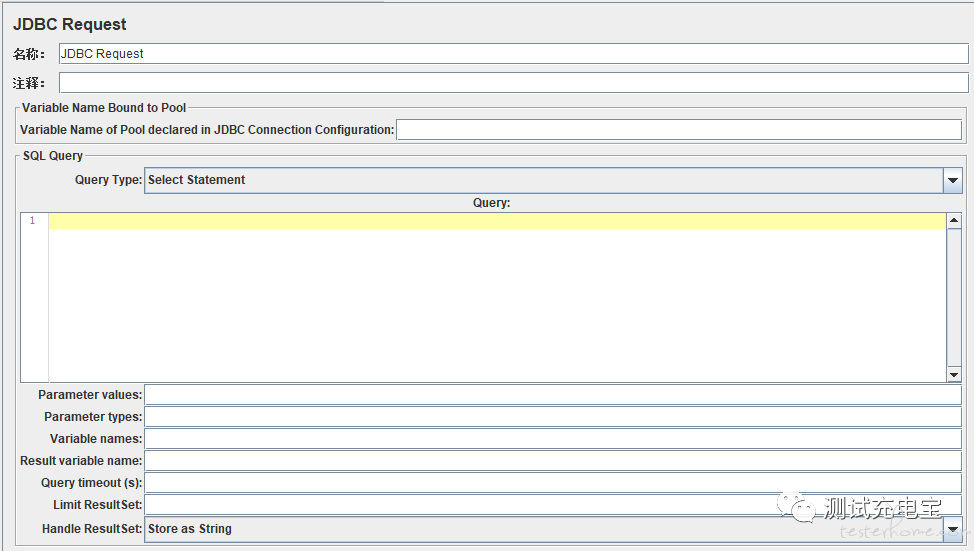
Database Connection Configuration

Database URL: jdbc:mysql://服务器地址:3306/数据库名

JDBC Driver class:数据库JDBC驱动类名：com.mysql.jdbc.Driver

Username:数据库连接用户名

password:数据库连接密码

[[](https://testerhome.com/uploads/photo/2020/63d08e30-4426-4b9b-833d-70f4ff6fda19.png!large)](https://testerhome.com/uploads/photo/2020/63d08e30-4426-4b9b-833d-70f4ff6fda19.png!large)

Variable Name Bound to Pool

Variable Name ....：数据库连接池的名字，需要与JDBC Connection Configuration的Variable Name Bound Pool名字保持一致

SQL Query

Query Type：此处支持方式多样，可以用于添加或者筛选数据，根据需要和Query配合使用

Query：填写的sql语句未尾不要加“;”

Parameter valus：参数值，顺序替代Query中的?；

Parameter types：参数类型

Variable names：保存sql语句返回结果的变量名 ，用于作为参数供调用

Result variable name：创建一个对象变量，保存所有返回的结果 ，供调用

Query timeout：查询超时时间

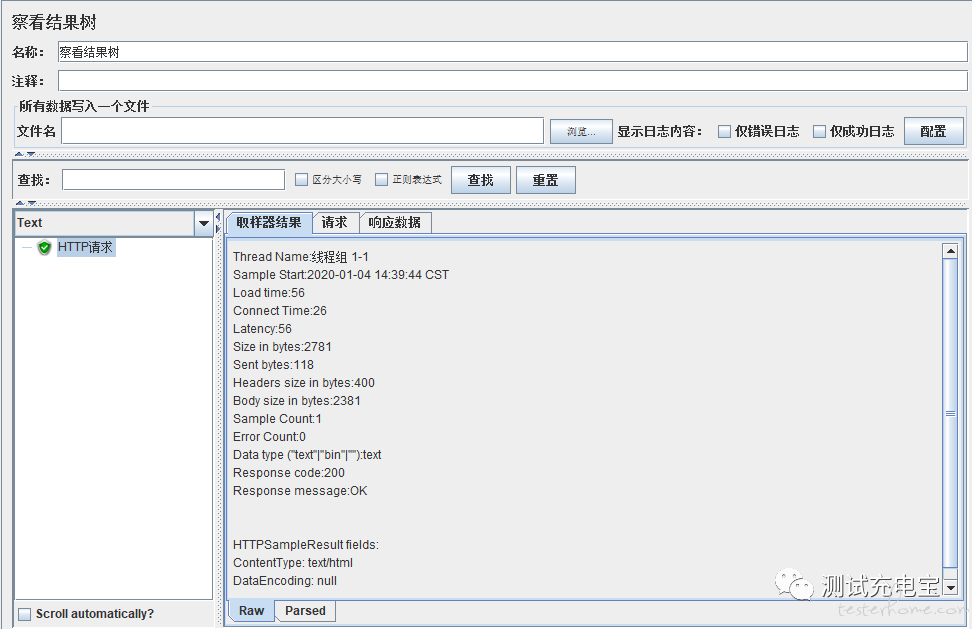
Handle result set：定义如何处理由callable statements语句返回的结果

JDBC Request取样器和JDBC Connection Configuration配置原件一起使用，前者主要用于连接数据库，前者用于执行sql语句保存变量供调用

**监听器**

重要元器件：查看结果树，聚合报告

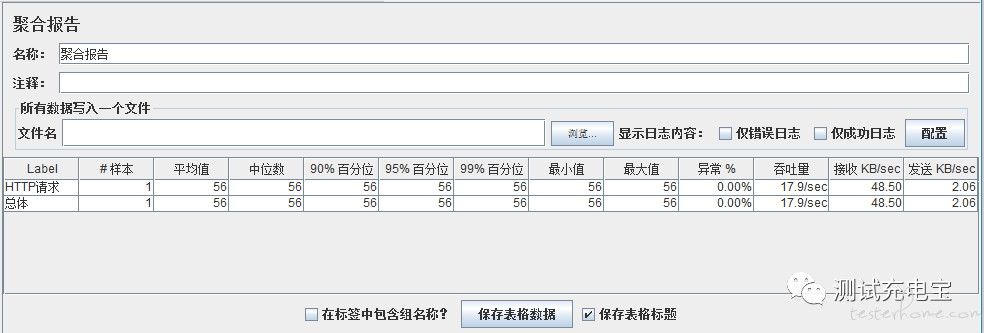
**查看结果树**

[[](https://testerhome.com/uploads/photo/2020/0f1750c7-0879-4720-9990-c84b8e91c567.png!large)](https://testerhome.com/uploads/photo/2020/0f1750c7-0879-4720-9990-c84b8e91c567.png!large)

左侧展示执行过的所有的取样器，成功显示绿色否则显示红色（没加断言绿色不一定代表成功）

右侧展示请求和响应的明细，有点类似浏览器F12Network下的详细信息或者Fiddler抓包详情

聚合报告

[[](https://testerhome.com/uploads/photo/2020/0fc5cb6b-a83f-4d1a-8828-1db1d9bf8f73.png!large)](https://testerhome.com/uploads/photo/2020/0fc5cb6b-a83f-4d1a-8828-1db1d9bf8f73.png!large)

统计每一个请求的请求数，平均值，最大，最小值，错误率，大约吞吐量(以请求数/秒为单位)和以kb/秒为单位的吞吐量

Label:请求的名称，就是我们在进行测试的httprequest sampler的名称

#样本:总共发给服务器的请求数量，如果模拟10个用户，每个用户迭代10次，那么总的请求数为：10\*10 =100次；

平均值:默认情况下是单个 Request 的平均响应时间，当使用了 Transaction Controller 时，以Transaction 为单位显示平均响应时间 ，单位是毫秒

中位数: 50%用户的请求的响应时间，中位数

90%百分位:90%的请求的响应时间

95%百分位:95%的请求的响应时间

99%百分位:99%的请求的响应时间

最小值:最小的响应时间

最大值:最大的响应时间

异常%:错误率=错误的请求的数量/请求的总数

吞吐量: 默认情况下表示每秒完成的请求数（Request per Second），当使用了 Transaction Controller 时，也可以表示类似 LoadRunner 的 Transaction per Second 数

接收KB/sec: 每秒从服务器端接收到的数据量

发送KB/sec: 每秒从服务器端接收到的数据量

特别说明：90% 百分位参数正确的含义

一组数由小到大进行排列，找到他的第90%个数（假如是3），那么这个数组中有90%的数将小于等于3 。

用在性能测试的响应时间也将非常有意义，也就是90%请求响应时间不会超过3 秒。

**总结**

本章节介绍的元器件比较多，我经常用到的是篇幅中介绍的监听器和配置元件下所有，以及HTTP请求。