

jmeter

jmeter-post请求自动变成get，提示请求方式错误，修改协议即可

修改协议：



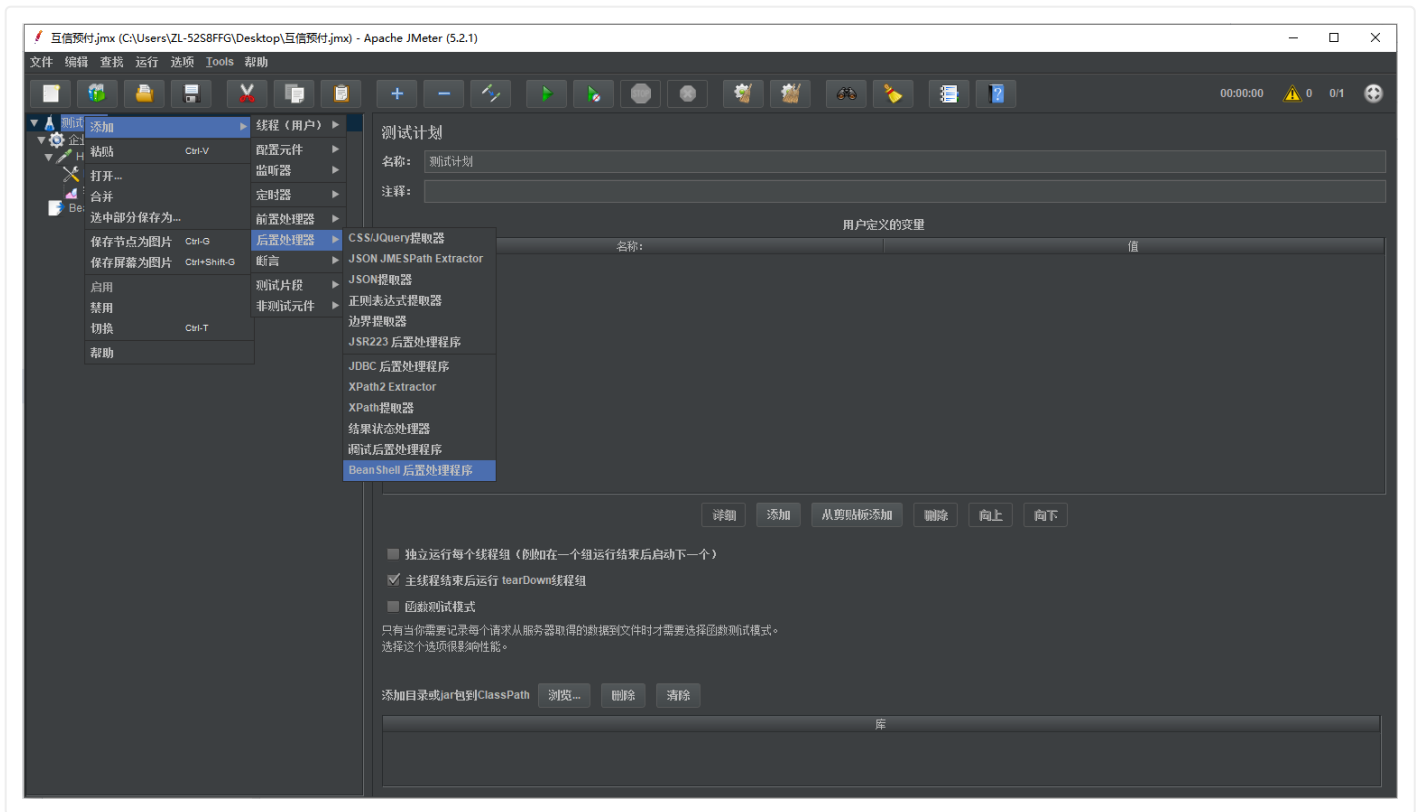
http 改为 https 或反之



Jmeter查看结果树 实现Response Body 为json格式化

在Jmeter的测试计划Test Plan 中添加 BeanShell PostProcessor;

路径：选中测试计划 > 添加 > 后置处理器 > BeanShell 后置处理程序 添加



在BeanShell PostProcessor 中的Script中插入代码

```
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;

try {

    String response_data = prev.getResponseDataAsString();

    log.info("response_data: " + response_data);
    ObjectMapper objectMapper = new ObjectMapper();

    Map readValue = objectMapper.readValue(response_data, Map.class);
    String writeValueAsString = objectMapper.writerWithDefaultPrettyPrinter().writeValueAsString(readValue);
    log.info("PrettyFromatJson result: " + writeValueAsString);

    prev.setResponseData(writeValueAsString);

} catch (JsonProcessingException e) {

    log.info("BeanShell PostProcessor failed=====
    ==", ex);

}
```

重启Jmeter
重启Jmeter，再次发送请求 查看是否可以；

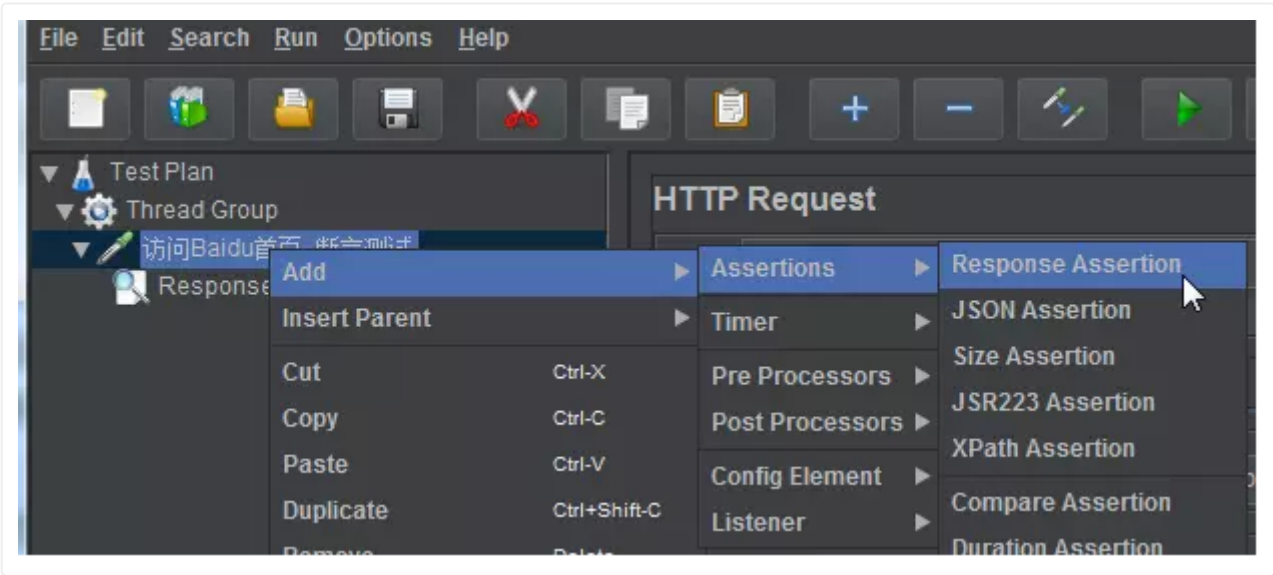
Response Body的中文乱码

注：我这边还出现个新问题，返回的Response Body 的数据格式正确了， 但是中间有的字段的内容是中文，导致输出的中文全部为"???",
解决方案：
找到Jmeter安装目录下的bin\jmeter.properties文件,将1075 行的"sampleresult.default.encoding"的值改为: "sampleresult.default.encoding=utf-8", 注意，要去掉"#"

jmeter常用四种断言

Response Assertion （响应断言）

- 1. 添加响应断言
对Web请求的响应结果进行验证



- 2. 输入需要匹配的字符串

模块类型	选项名称	配置说明
	Name	Response Assertion名称
	Comments	注释
	Apply to	断言应用的范围
	Main sample and sub-samples	作用于父节点取样器及其子节点取样器
	Main sample	仅作用于父节点取样器

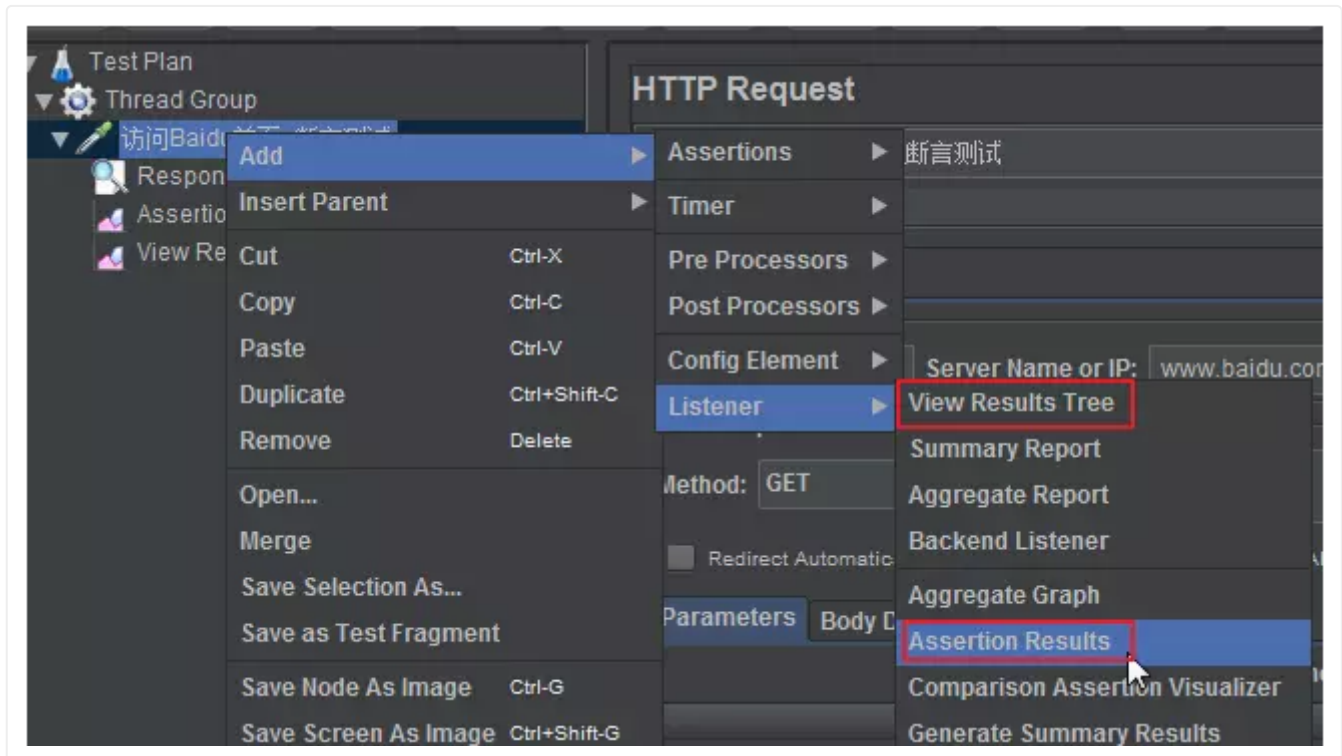
only	
Sub-samples only	仅作用于子节点取样器
Jmeter Variable Name to use	作用于Jmeter变量（输入框中可输入Jmeter的变量名称）
Field to Test	测试的字段
Text Response	匹配从服务器返回的响应文本（不包括Response Headers）
Response Code	匹配响应状态码
Response Message	匹配响应信息。如：OK
Response Headers	匹配响应头
Request Headers	匹配请求头
URL Sampled	匹配URL链接
Document(text)	匹配文档内容
Ignore Status	一个请求多项响应断言时，忽略某一项断言的响应结果，而继续下一项断言
Request Data	匹配请求数据
Pattern Mactching Rules	匹配的规则
Contains	返回的结果包括所指定的内容，支持正则匹配
Matches	根据指定内容进行匹配
Equals	返回结果与所指定的内容一致
Substring	返回结果包括所指定结果的字符串，不支持正则匹配
Not	不进行匹配就算是Pass
Or	暂不确定该模式的用法
Patterns to	

Test

Patterns to Test

需要匹配的正则表达式、字符串。可以添加多项，每一项会分开进行验证，若某一项验证失败，则其后的不会再进行验证。

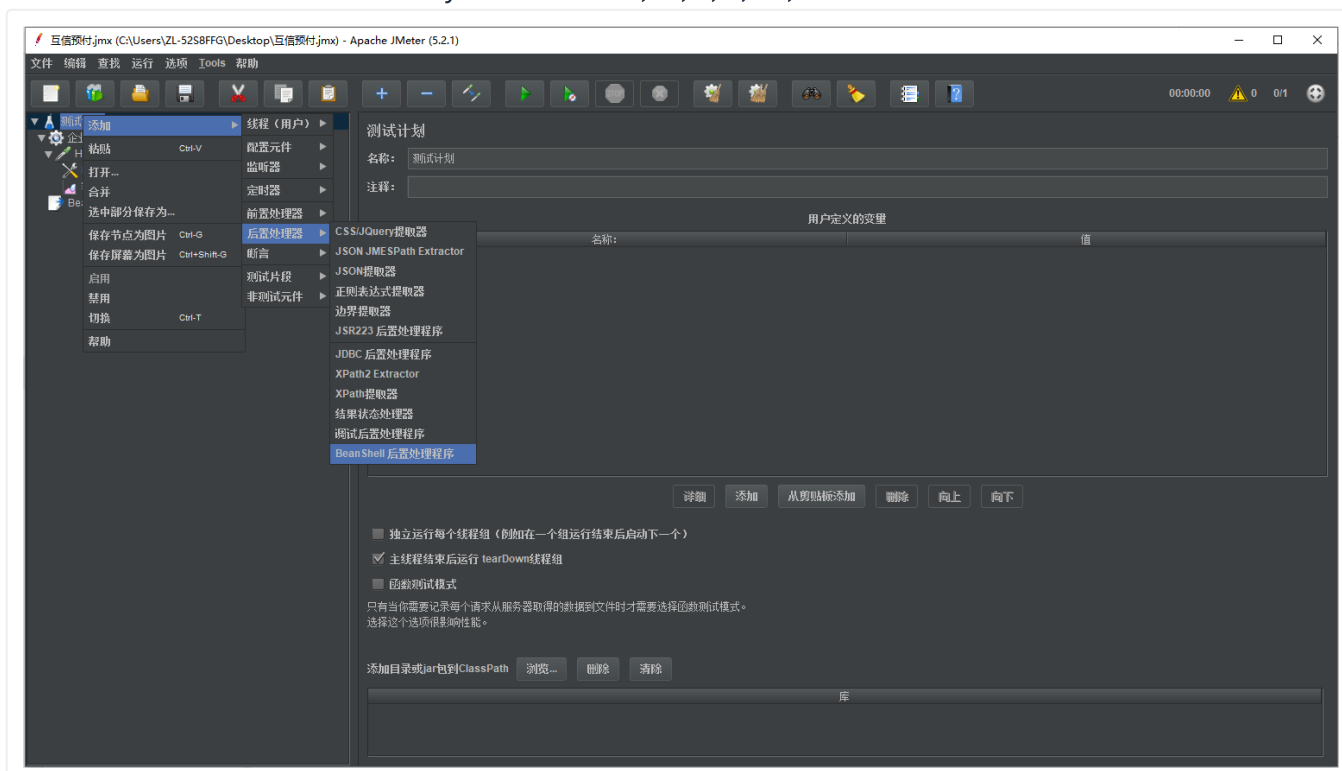
3. 添加：断言结果(Assertion Results)、查看结果树(View Results Tree)

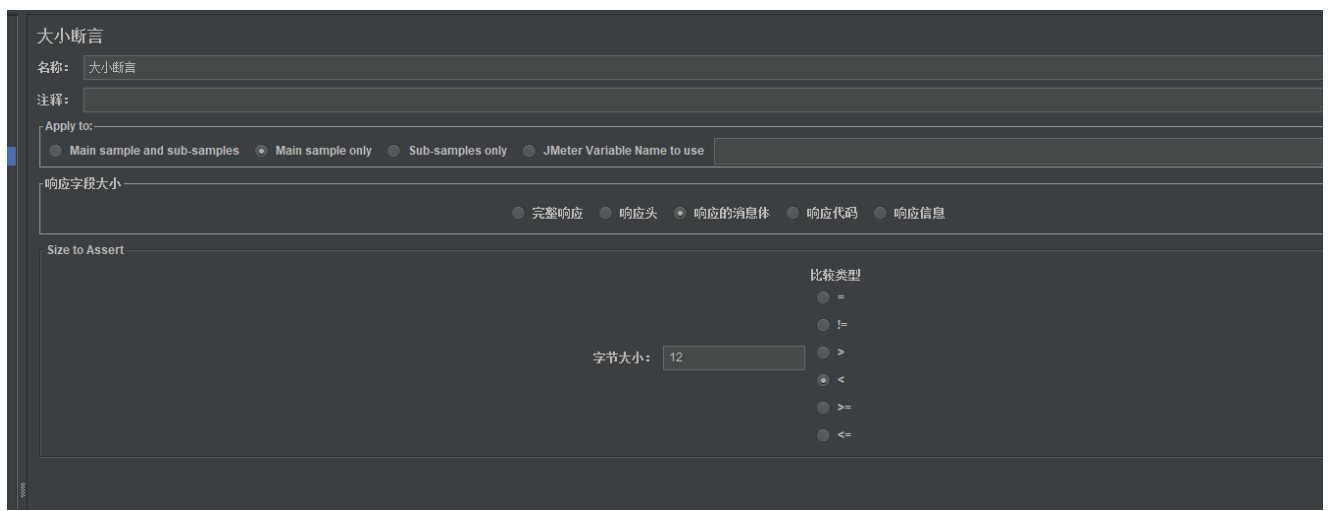


4. 运行Test Plan中的线程组，进行断言检查

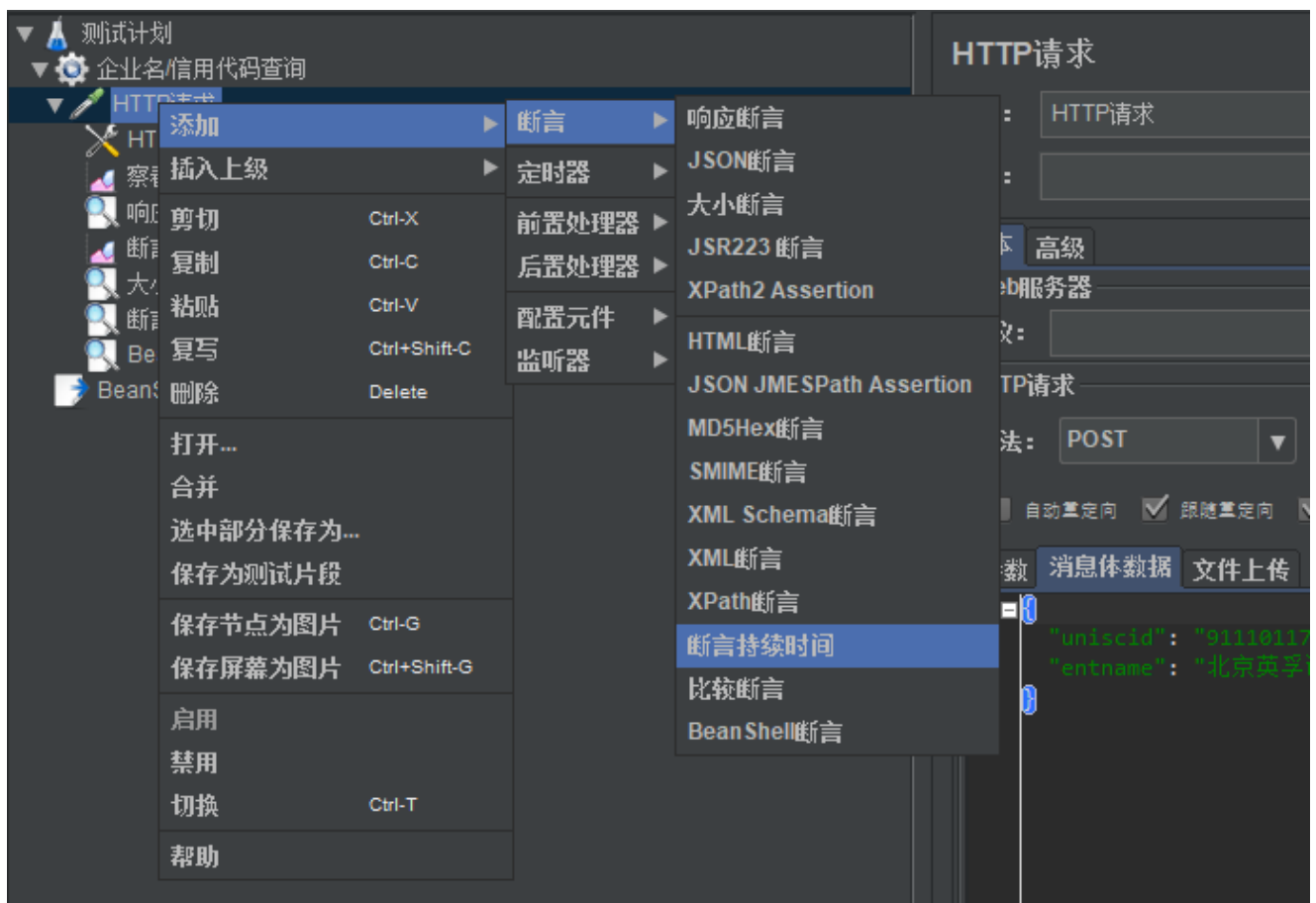
Size Assertion（数据包字节大小断言）

判断响应结果是否包含正确数量的byte。可定义 (=, !=, >, <, >=, <=)



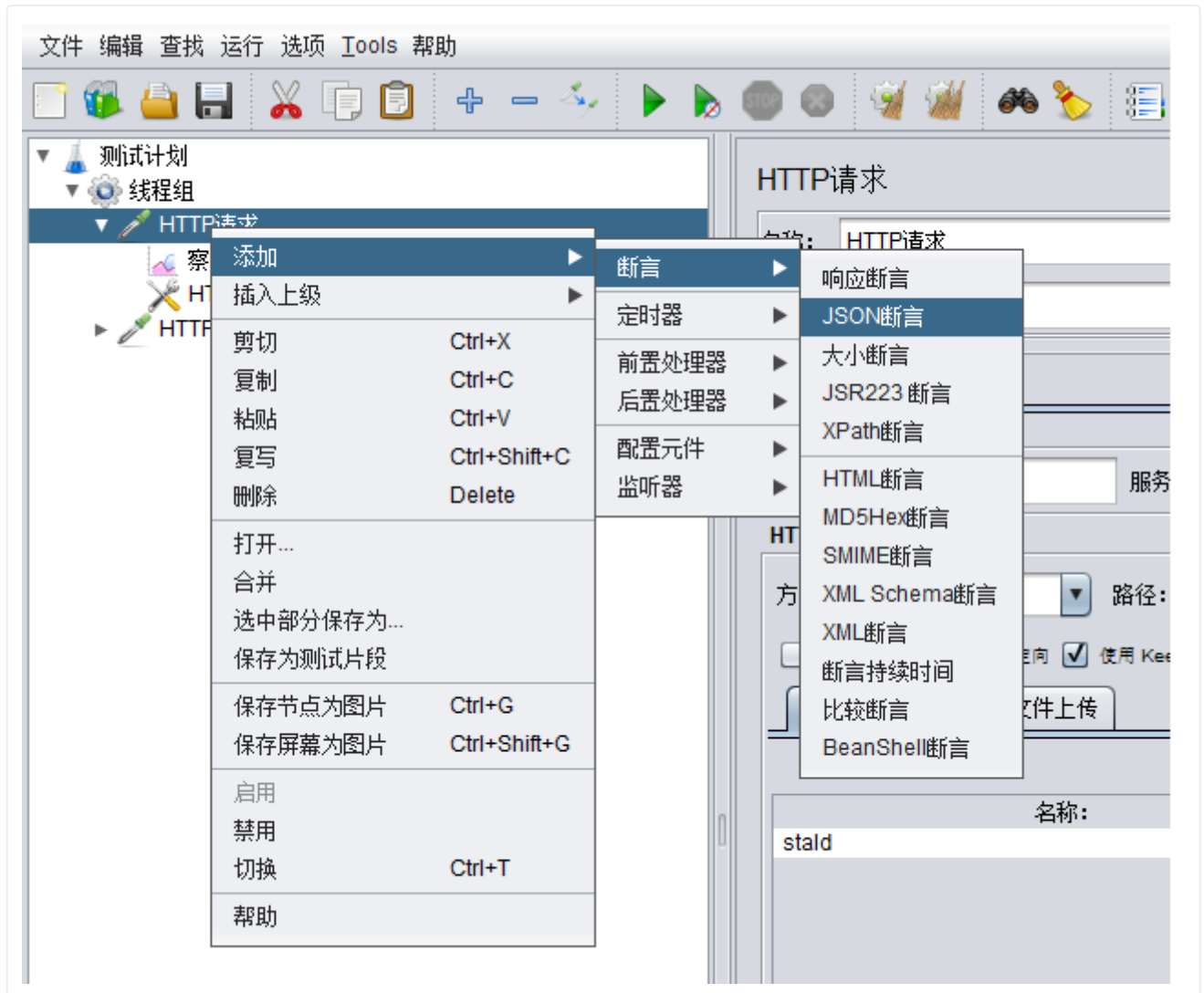


Duration Assertion（持续时间断言）

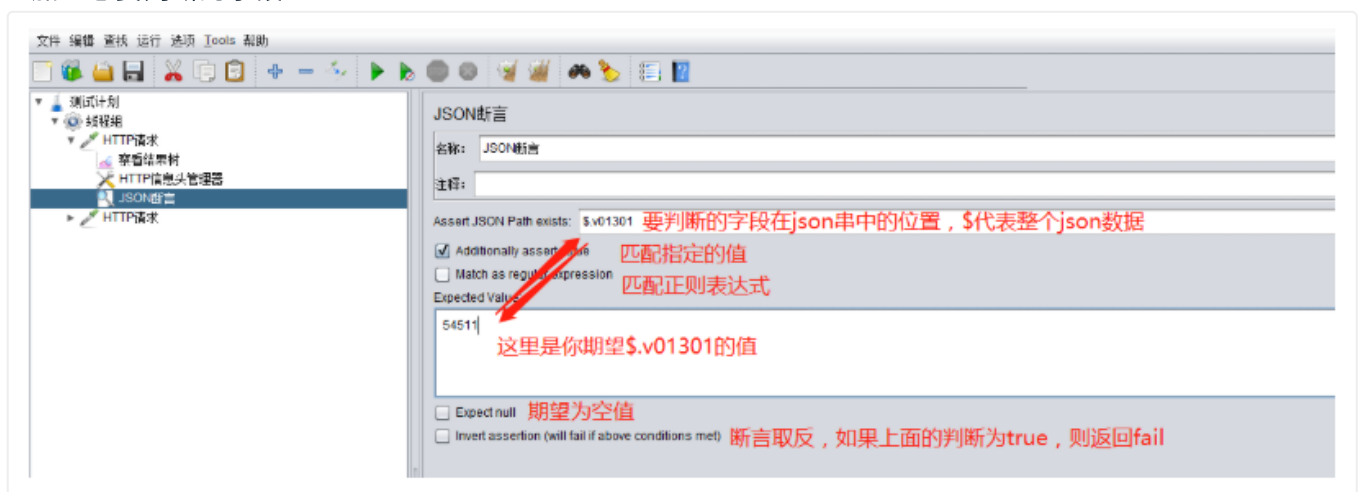


json断言

1.添加一个Json断言



2.加上想要判断的字段



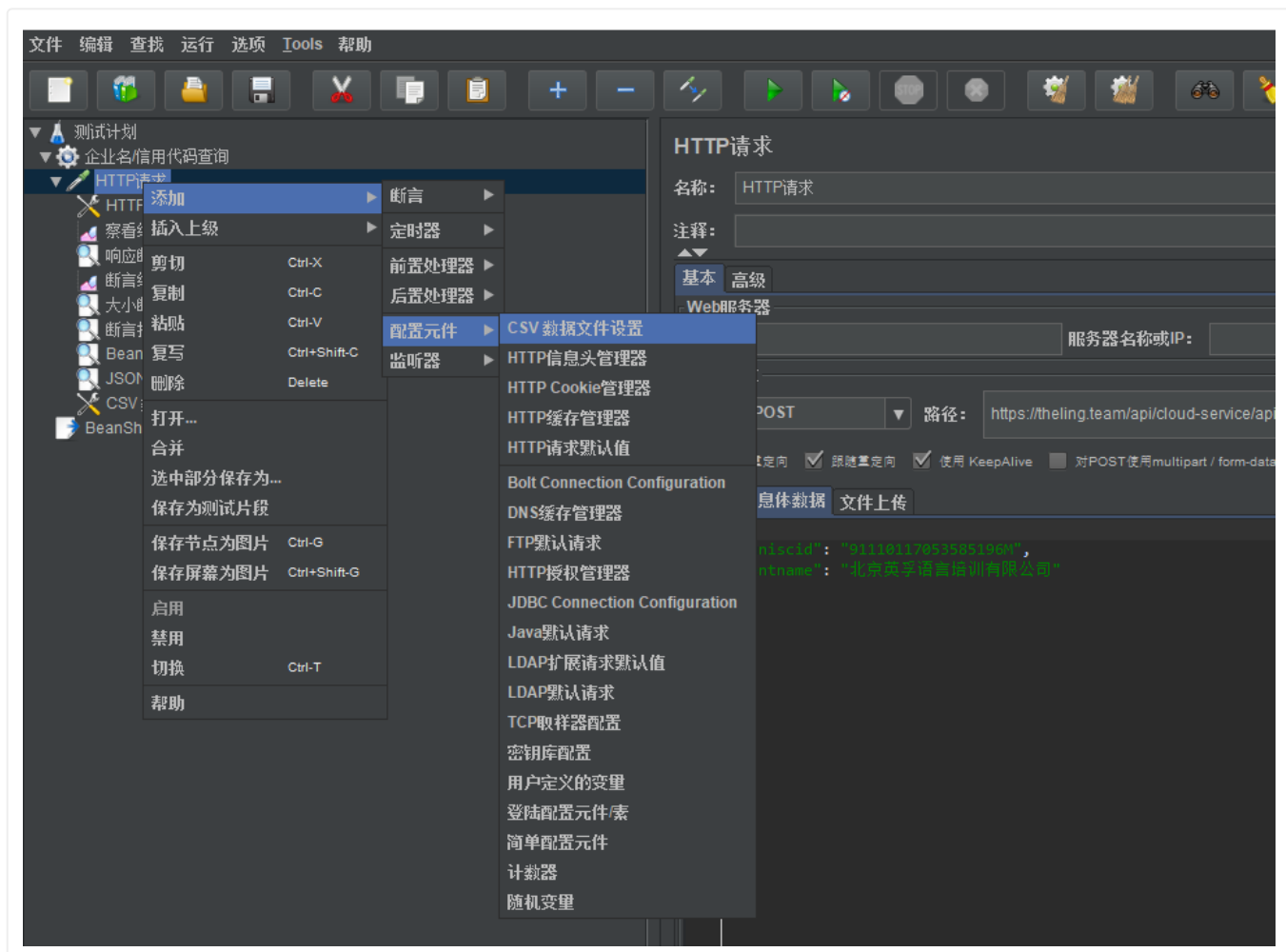
参数化

从文件读取

1.新建1个csv文件，里面写上请求参数

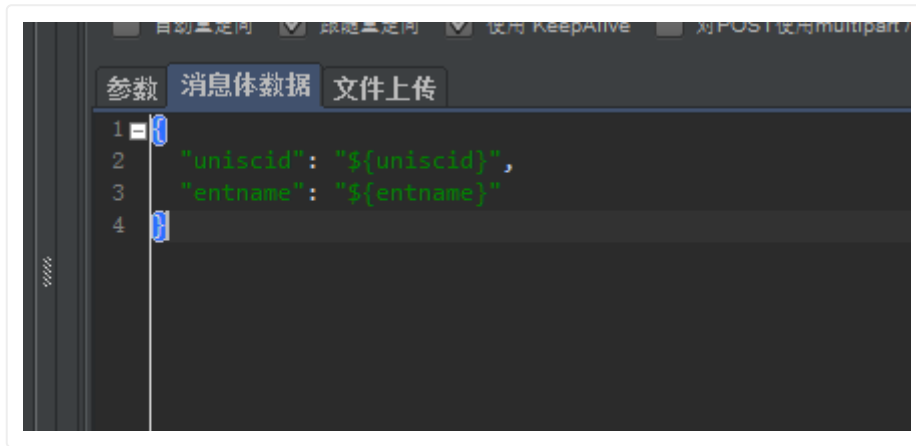
	A	B	C	D
1	uniscid	entname		
2	123456789012345678	11		
3	123456789012345678	aaaaa		
4	123456789012345678	@!\$\$\$!\$#		
5	123456789012345678	AAA		
6		北京英孚语言培训有限公司		
7	91110117053585196M			
8		北京英孚语言培训有限公司		
9	91110117053585196M			
10				
11				
12				
13				
14				
15				

2.添加CSV数据文件设置





3.在请求数据的值中用`${uniscid}`、`${entname}`来获取文件中的值



4.线程组设置循环数，循环读取csv中的数据

线程组

名称: 企业名/信用代码查询

注释:

在取样器错误后要执行的动作

☒ 继续
 ☐ 启动下一进程循环
 ☐ 停止线程
 ☐ 停止测试
 ☐ 立即停止测试

线程属性

线程数: 1

Ramp-Up时间(秒): 1

循环次数 ☐ 永远 3☒ Same user on each iteration☐ 延迟创建线程直到需要☐ 调度器

持续时间(秒)

启动延迟(秒)

Text

☒ HTTP请求☐ HTTP请求☐ HTTP请求

取样器结果 请求 响应数据

Request Body

Request Headers

```

1 POST https://theling.team/api/cloud-service/api/enterp
2
3 POST data:
4 {
5   "uniscid": "91110117053585196M",
6   "entname": "北京英孚语言培训有限公司"
7 }
8
9 [no cookies]
10

```

用户定义的变量

如果是固定不变的一些配置项，不需要多个值的时候，也可以使用用户已定义的变量
添加用户定义的变量

测试计划

企业名/信用代码查询

HTTP请求

HTTP信息头管理器

察看结果树

断言结果

JSON断言

CSV数据文件设置

BeanShell 监听器

用户定义的变量

BeanShell 后置处理器

用户定义的变量

名称: 用户定义的变量

注释:

用户定义的变量

名称	值	描述
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzU1NiJ9.eyJleHAiOiJlbnV9.eyJleHAiOiJlbnV9.eyJleHAiOiJlbnV9...	

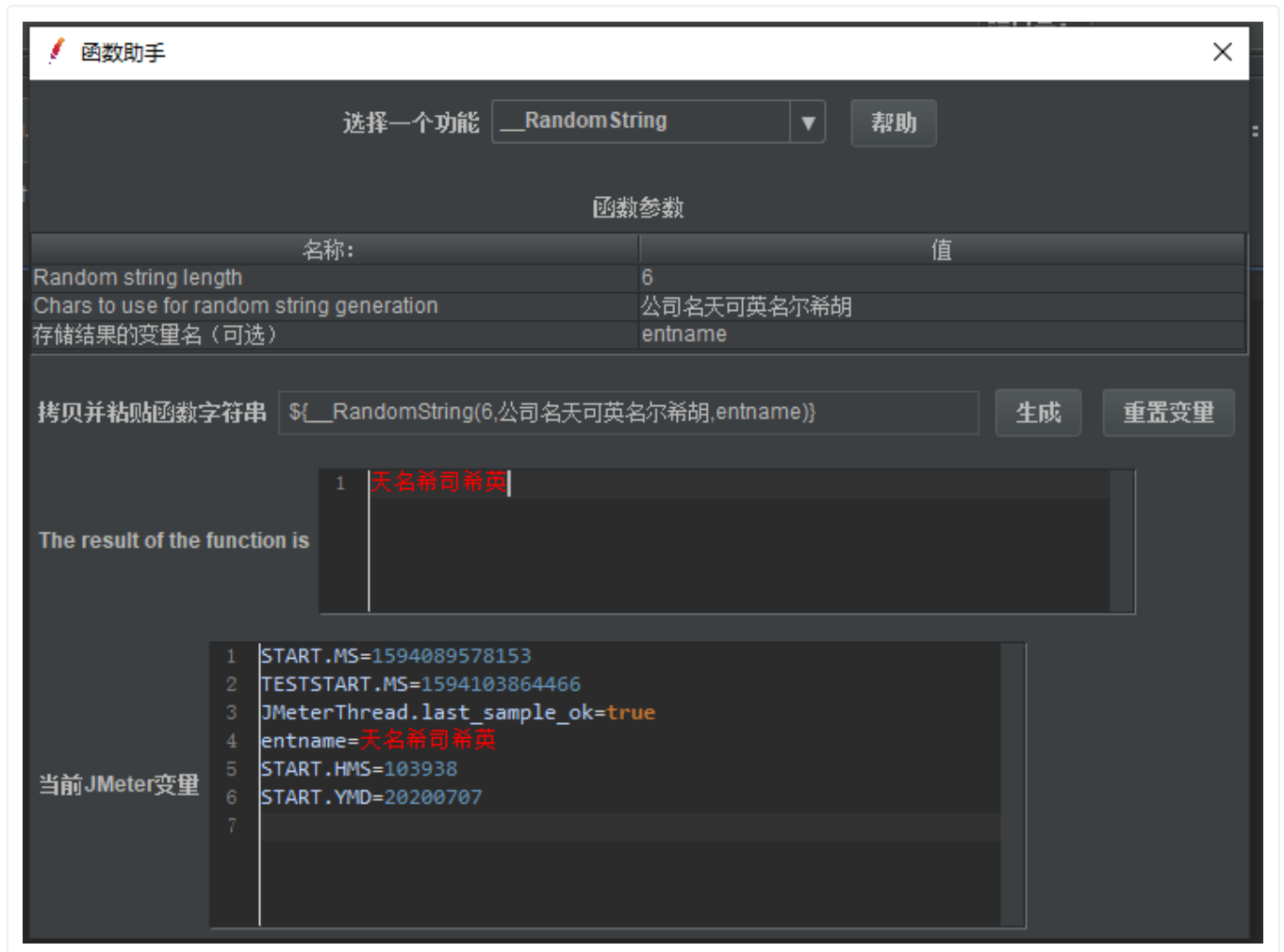
引用变量



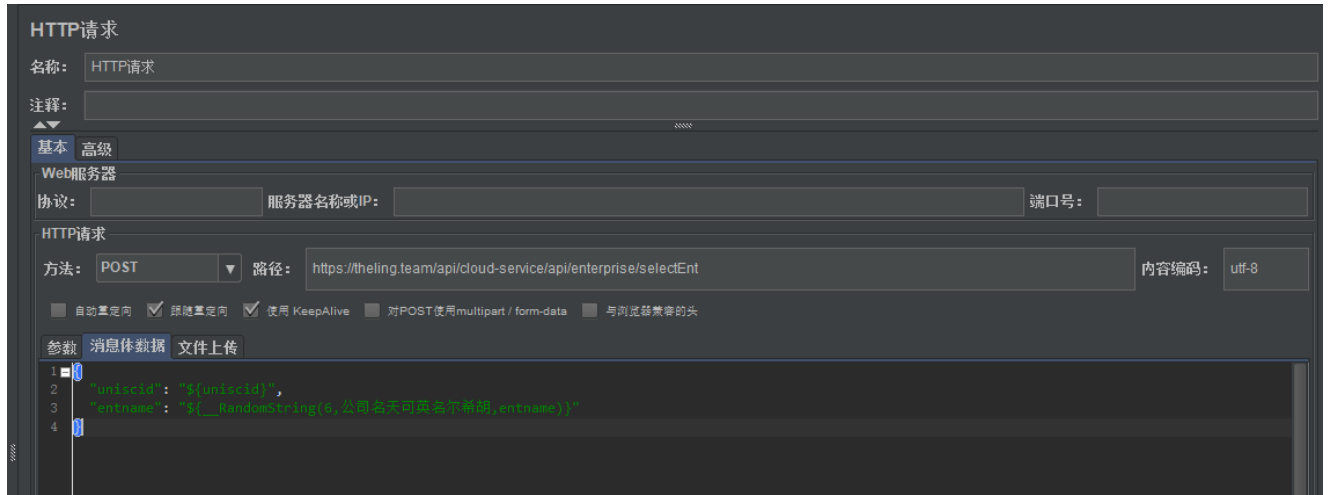
生成随机字符串

使用jmeter自带函数

点击：Tools->函数助手对话框，打开函数助手，设置长度，存储名称等



在请求参数中引用



jmeter BeanShell PreProcessor 写脚本生成

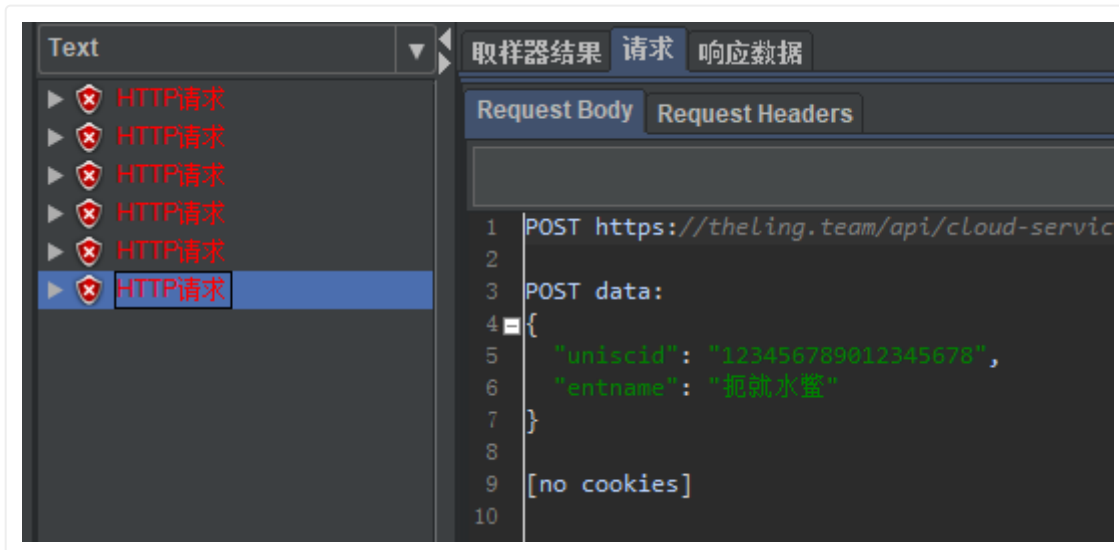
- 1.右键-添加-前置处理器-添加 BeanShell PreProcessor
- 2.script 栏添加脚本代码

```
import java.util.Random;

public class Random_str {
    public static String RandomJianHan(int len) {
        String ret = "";
        for (int i = 0; i < len; i++) {
            String str = null;
            int hightPos, lowPos; // 定义高低位

            Random random = new Random();
            random.setSeed(System.currentTimeMillis());
            hightPos = (176 + Math.abs(random.nextInt(39))); // 获取高位值
            lowPos = (161 + Math.abs(random.nextInt(93))); // 获取低位值
            byte[] b = new byte[2];
            b[0] = (new Integer(hightPos).byteValue());
            b[1] = (new Integer(lowPos).byteValue());
            try {
                str = new String(b, "GBk"); // 转成中文
            } catch (UnsupportedEncodingException ex) {
                ex.printStackTrace();
            }
            ret += str;
        }
        return ret;
    }
}
```

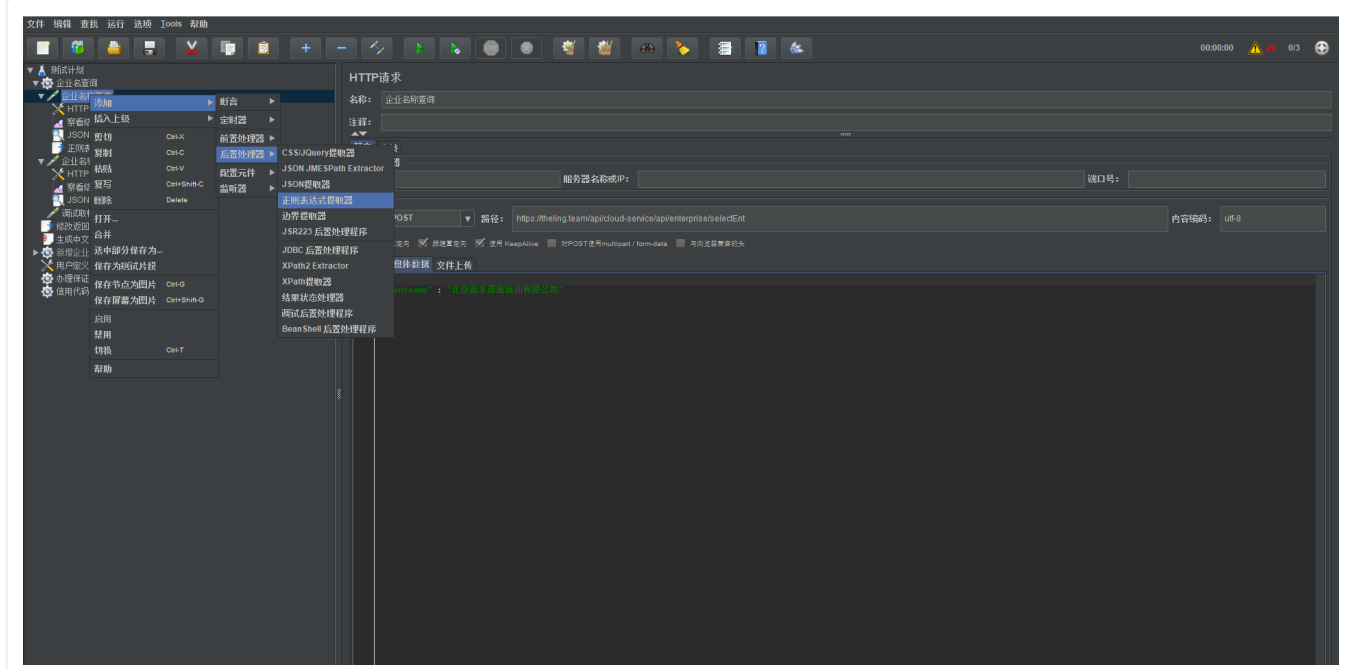
```
Random_str ran = new Random_str();  
String content1 = ran.RandomJianHan(4); //此处生成的是长度为4的字符串  
vars.put("content_post",content1);
```



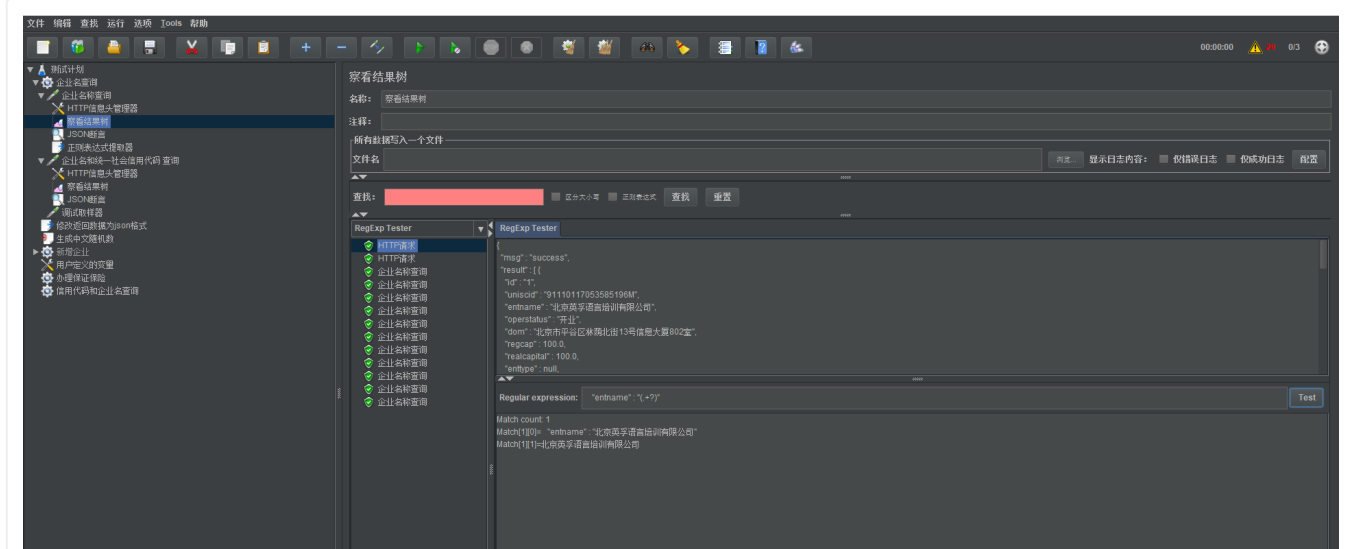
关联

正则表达式提取器

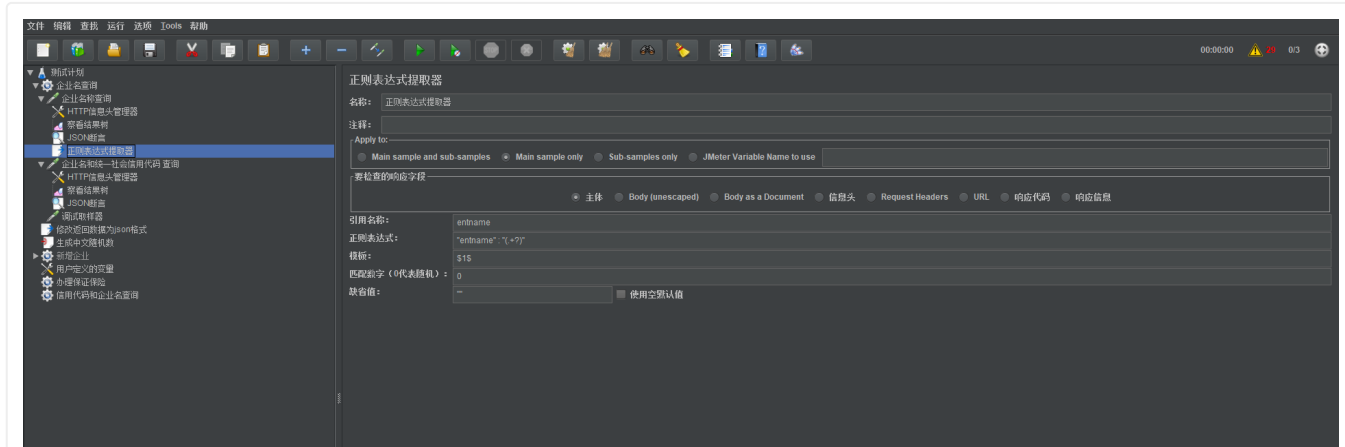
1. 在提取参数的取样器中添加:正则表达式提取器



在原来的取样器的查看结果树中 写出自己想要提取参数的正则表达式



2. 将正则表达式 复制到正则表达式提取器中



APPLY to:作用范围（返回内容的断言范围）

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

Main sample only: 仅作用于父节点的取样器

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)

要检查的响应字段：需要检查的响应报文的范围

主体：响应报文的主体

Body(unesaped):主体，响应的主体内容且替换了所有的html转义符，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

Response Headers：响应信息头

Request Headers:请求信息头

URL：统一资源定位符，即Internet上用来描述信息资源的字符串

响应代码:响应状态码，比如200、404

响应信息:响应信息

引用名称（Reference Name）：Jmeter变量的名称，存储提取的结果；即下个请求需要引用的值、字段、变量名（例子中我提取的是SOCIAL_NO）

正则表达式（Regular Expression）：使用正则表达式解析响应结果

模板：代表从正则表达式结果引用的样式，其实结果是一组，而不是一个。\$0\$代表这一组结果的全部，\$1\$代表这一组结果的第1个，以此类推；\$1\$2\$代表该正则表达式一组结果中的第1个和第2个，俩结果挨在一起中间没有间隔；\$3\$,\$4\$代表该正则表达式一组结果中的第3个和第4个，俩结果间有一个逗号相连。

匹配数字：0代表正则表达式结果组中随机，1代表全部。

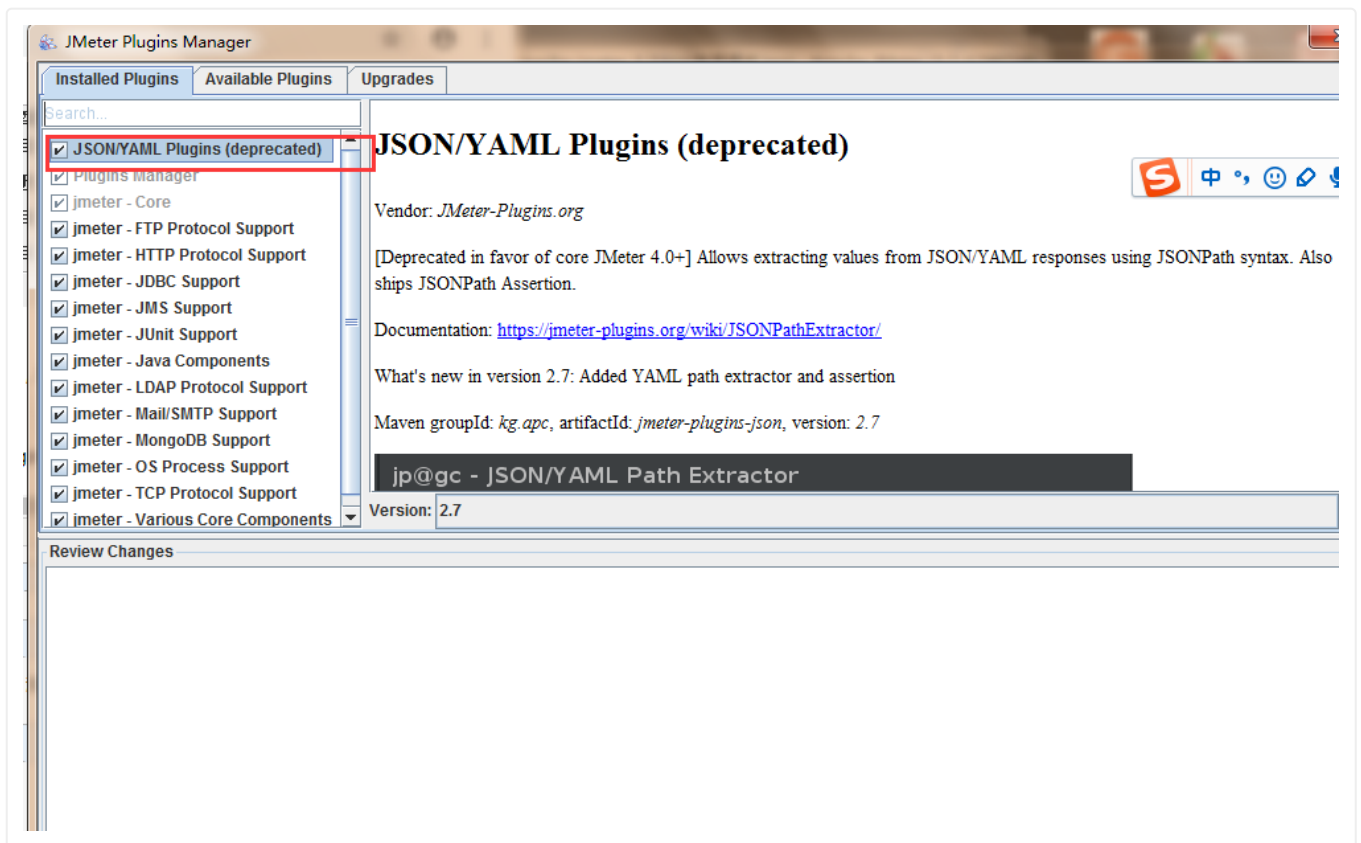
缺省值：当引用不对时显示传递的信息。

1. 引用提取的参数

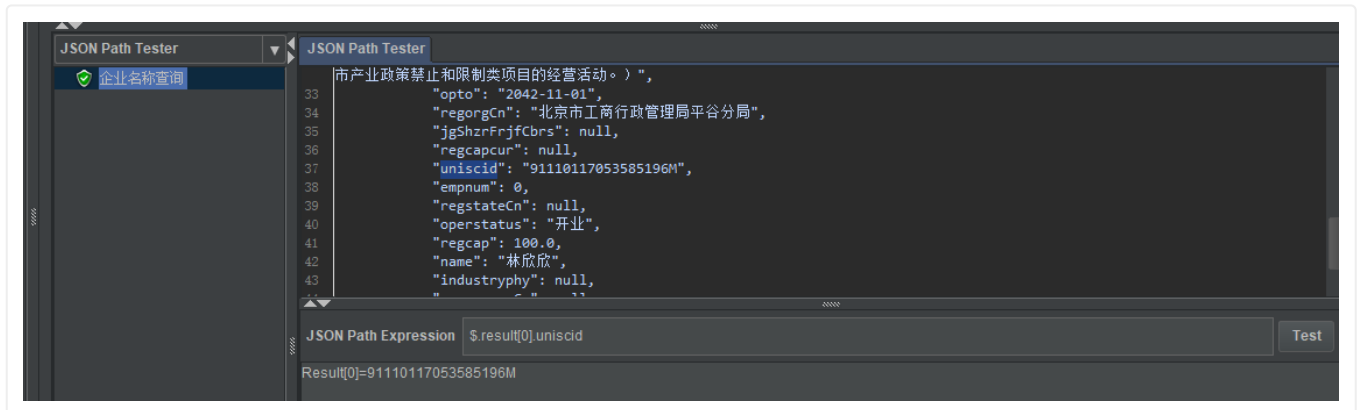


jp@gc - JSON Path Extractor提取器

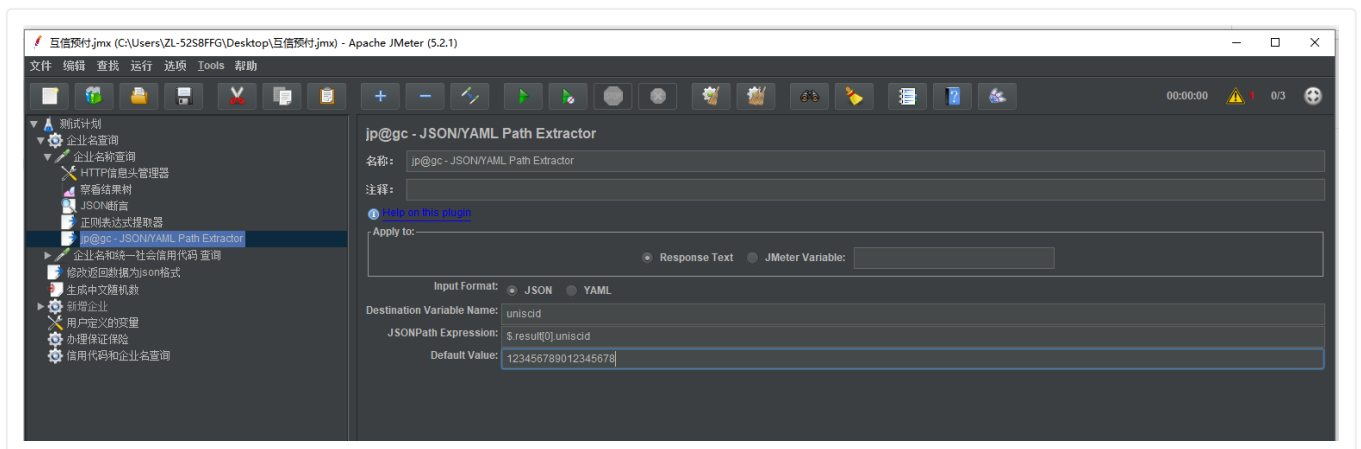
- 1、前提：先打开jmeter，看看后置处理器中是否已经存在JSON Path Extractor，如果没有再进行第二步操作。
- 2、下载plugins-manager.jar并将其放入lib / ext目录，然后重启JMeter。
下载地址：<https://jmeter-plugins.org/install/Install/>
- 3、安装插件 options->plugins manager>Available Plugins>勾选JSON/YAML Plugins(deprecated)>点击【Apply Changes and Restart JMeter】
- 4、安装完成如下图：



5、获取正确的json字段路径



6、输入关联变量名称

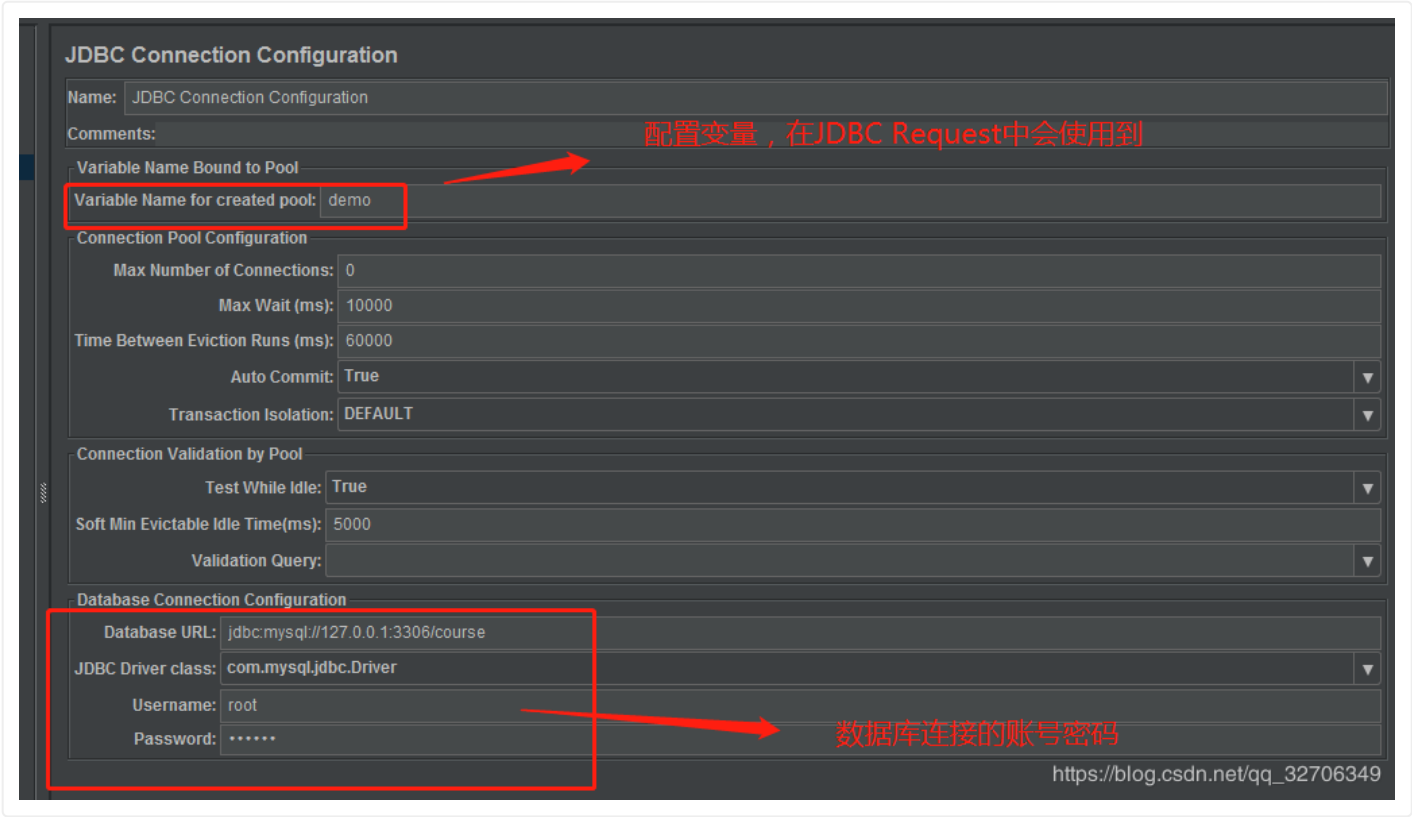


7、在后面的http请求中获取关联变量的值

数据库测试

1、需将mysql-connector-java-X.X.X-bin.jar放入到jmeter安装目录下的./lib目录，重启jmeter

2、配置JDBC Connection Configuration



其中：

Database URL为数据库的JDBC连接字符串

JDBC Driver class 为驱动类名称，需将对应jar包放入到jmeter's/lib目录下

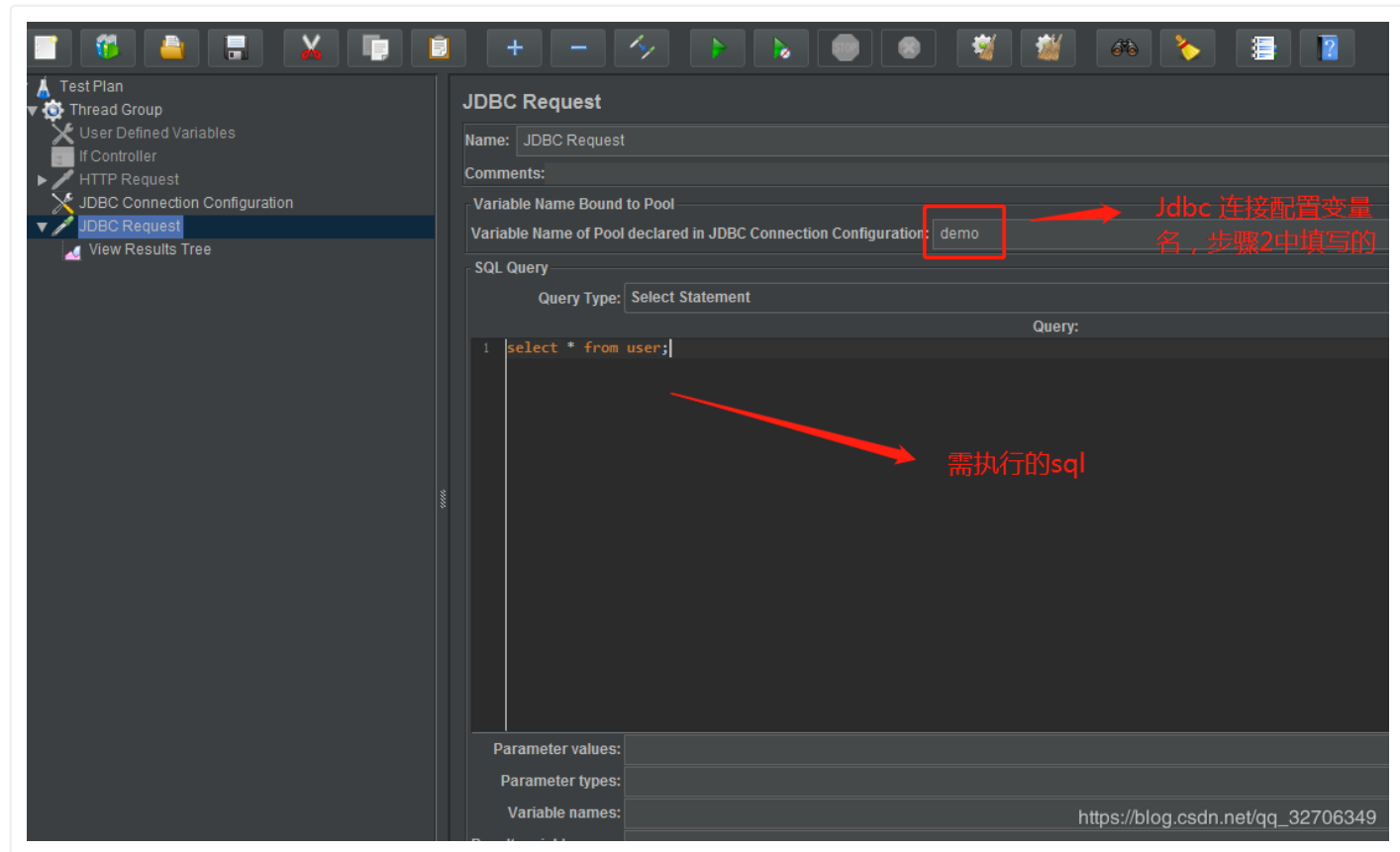
Username :数据库用户名

Password:数据库账号对应的密码

以下是各数据库Database URL、JDBC Driver class 填写方式

数据库名	Database URL	
MySQL	jdbc:mysql://host[:port]/dbname	
PostgreSQL	jdbc:postgresql:{dbname}	
Oracle	jdbc:oracle:thin:@//host:port/service OR jdbc:oracle:thin:@(description=(address=(host={mc-name})(protocol=tcp)(port={port-no}))(connect_data=(sid={sid})))	
Microsoft SQL Server (MS JDBC driver)	jdbc:sqlserver://host:port;DatabaseName=dbname	com.micros

3、配置JDBC Request



4、在JDBC Request中写sql语句，注意：如果使用参数值的形式，Query Type一定要选择Prepared Select Statement，否则会报错

